知识蒸馏专题笔记

贾配洋 2019.01.16

0.灵感来源

蒸馏神经网络 2014 年由 Hinton[1]提出，故事要从昆虫记中昆虫繁殖讲起。“**蝴蝶以毛毛虫的形式吃树叶积攒能量逐渐成长，最后变换成蝴蝶这一终极形态来完成繁殖**。”同一个体，在面对不同环境、不同任务时，个体形态却不同。不同形态是为了完成特异性任务而产生的变化，从而使个体能够更好适应新环境。比如毛毛虫形态是为了更方便吃树叶，积攒能量，但为了增大活动范围提高繁殖率，毛毛虫要变成蝴蝶完成繁殖。

蒸馏神经网络，本质上就是要完成一个从毛毛虫到蝴蝶的转变。因为在使用神经网络时，训练时模型和实际应用模型往往是相同的，就好像一直是一个毛毛虫，既吃树叶积累能量，又做繁殖任务，既臃肿又效率低下。使用同样形态模型，一方面会导致模型不能针对特定任务快速学习，另一方面实际应用中用训练时非常庞大的模型会造成使用开销负担过重。

1.基本思想

1.1 什么是知识蒸馏?

观点 1：蒸馏神经网络更接近于**迁移学习**（Transfer Learning [2]），目的是将庞大网络学到知识转移到小网络模型，即不改变网络复杂度下，通过增加监督信息丰富程度来提升性能。

关键点：知识获取 知识转移

观点 2：知识蒸馏是一种**模型压缩方法**，用于模型压缩指的是在 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（Model Compression[3]角度理解）。可提供 student 在 one-hot label 上学不到的 soft label 信息，包含了类别间信息，以及 student 小网络学不到而 teacher 网络可以学到的特征表示‘知识’，所以一般可提高 student 网络精度。

1.2 常见集中思想

(1)**softmax 层输入比类别标签包含更多监督信息**，使用 logistics 代替类别标签对小模型进行训练，将小模型训练转化为了回归问题。让小模型输出尽量接近大模型的 logits。因为小模型隐层要足够宽，所以参数没有明显减少，效果有限。

(2)softmax 输出层包含了每个类别概率，包含了更多信息，用超参数控制预测概率的平衡程度。**最终损失函数由小模型预测结果和大模型 logistics 的交叉熵，和小模型预测结果和类别标签的交叉熵组成。**通过调节权重确定两部分重要程度。但当类别较多时模型难收敛，因为与维度紧密相关。与 logits 相比，前一层输出包含了更多噪声和无关信息。因此先取出无关维度（保留足够强区分维度，维度间低相关）。效果会提高。

1.3 紧凑网络结构(可忽略之)

(1)挤压

维度不高表达不强，维度高了参数增多，容量与参数的平衡用 1*1 的卷积进行降维，得到多通道信息，特征紧密，保证模型泛化;

(2)扩张

为了减少参数，部分使用 1*1 代替大的卷积核，但为了保证不同核输出拼接完整，要对大的卷积输入进行合适的填充像素;

(3)Squeezenet:

三条卷积操作，扩张卷积，反卷积，普通卷积，然后合并输入下一层。实现了 4.1M 参数和 googlenet 效果一样。

2.论文解读

Distilling the knowledge in a neural network, Hinton, NIPS 2014

Hinton 这篇论文数学推导严谨，并通过巧妙实验来验证了可行性。

2.1 Abstract

提高几乎所有机器学习算法性能的一种非常简单的方法是**在相同数据上训练许多不同的模型，然后对它们的预测进行平**均。**缺点**：集成模型预测非常麻烦且**计算量过大而不能允许部署到大量用户系统**，尤其各模型都是大网络。已证明，有可能将集成模型中的知识压缩到一个更易于部署的单一模型中，并且我们使用不同压缩技术进一步开发这种方法。

我们在 MNIST 上取得了一些惊人成果，将集成模型中的知识提炼成单一模型，显着改善被大量使用的商业系统中的声学模型。我们还介绍了一种由一个或多个完整模型和许多专业模型组成的新型集合，它们学会区分细粒度细粒度的类别，而且这些类别是完整模型不能区分的。与专家混合不同，这些专业模型可以快速并行地进行训练。

在用神经网络训练大规模数据集时，为了处理复杂的数据分布：

(1)**建立复杂神经网络模型**，例如上百层残差网络，往往含有多达几百万个参数；(2)**混合多种模型**，将几个大规模神经网络在同一个数据集上训练好，然后综合（ensemble）多个模型，得到最终分类结果。但复杂模型在新场景下重新训练成本过高，模型过于庞大而难以大规模部署（deployment）。所以，最基本想法就是将大模型学习出来的知识作为先验，**将先验知识传递到小规模神经网络**中，之后实际应用中部署小规模神经网络。三点依据：

(1)大规模神经网络得到的**类别预测包含了数据结构间的相似性**；

(2)有了先验的小规模神经网络**只需要很少的新场景数据就能够收敛**；

(3)Softmax 函数随着温度变量（temperature）的升高分布更均匀。

神经网络模型在预测最终分类结果时，往往通过 softmax 函数产生概率分布：

$$q_i = \frac{exp(z_i/T)}{\sum_j exp(z_j/T)}$$

T 为温度参数，是一个超参数，q_i 是 i 类概率值大小。T 越高，产生概率输出约 soft。

2.2 数据结构间相似性如何理解？

一个大规模网络如 ImageNet，能预测上千种类别，正确类别概率值能够达到 0.9，错误类概率值可能分布在 10^-8~10^-3 区间。虽然每个错误类别概率值都很小，但是 10^-3 还是比 10^-8 高了五个数量级，这也反映了数据之间的相似性。

比如一只狗，在猫这个类别下概率值可能是 0.001，而在汽车类别下概率值可能只有 0.0000001 不到，这能反映狗和猫比狗和汽车更相似，这就是大规模神经网络能够得到更为丰富数据结构间相似信息。

2.3 将大规模神经网络 soft target 作为训练目标

由于大规模神经网络在训练时虽然是通过 0-1 编码来训练，由于最后一层往往使用

softmax 层来产生概率分布，所以这个概率分布其实是一个比原来的 0-1 编码硬目标（hard target）更软的软目标（soft target）。这个分布是由很多（0,1）之间的数值组成的。

Using soft targets instead of hard targets:

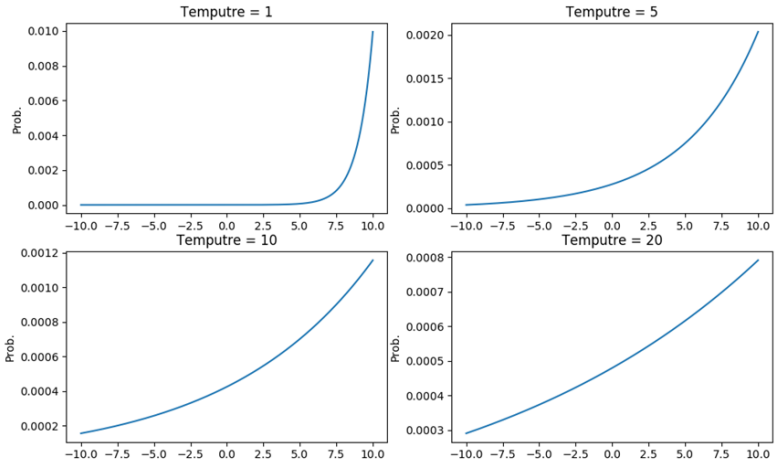
- A lot of helpful information can be carried in soft targets that could not possibly be encoded with a single hard target.
- Using far less data to fit the 85M parameters of the baseline speech model: Soft targets allow a new model to generalize well from only 3% of the training set.

System & training set	Train Frame Accuracy	Test Frame Accuracy
Baseline (100% of training set)	63.4%	58.9%
Baseline (3% of training set)	67.3%	44.5%
Soft Targets (3% of training set)	65.4%	57.0%

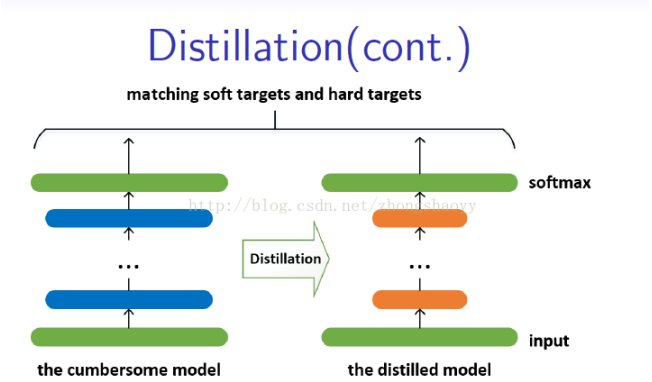
Using soft targets to prevent specialists from overfitting:

- If using a full softmax over all classes for specialists: soft targets may be a much better way to prevent them overfitting than using early stopping
- If a specialist is initialized with the weights of the generalist, we can make it retain nearly all of its knowledge about the non-special classes by training it with soft targets for the non-special classes in addition to training it with hard targets.

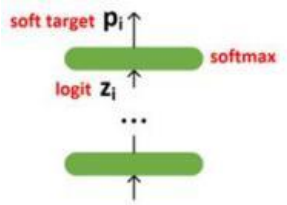
同一个样本，用在大规模神经网络上产生的软目标来训练一个小的网络时，因为并不是直接标注的一个硬目标，学习起来会更快收敛。更巧妙的是，这个样本我们**甚至可以使用无标注数据来训练小网络，因为大的神经网络将数据结构信息学习保存起来，小网络就可以直接从得到的 soft target 中来获得知识**。这个做法类似学习了样本空间嵌入（embedding）信息，从而利用空间嵌入信息学习新网络。按照 softmax 的分布来看，随着 **T 参数(温度)增大，这个软目标分布更加均匀**。因此：



(1)首先用较大 T 值来训练模型，复杂神经网络能够产生更均匀分布的软目标；
(2)小规模神经网络用相同 T 值来学习由大规模神经产生的软目标，接近这个软目标从而学习到数据的结构分布特征；
(3)最后实际应用，将 T 值恢复到 1，让类别概率偏向正确类别。
所以，蒸馏神经网络取名为蒸馏（Distill），其实是一个非常形象的过程。
我们把数据结构信息和数据本身当作一个混合物，分布信息通过概率分布被分离出来。首先，T 值很大，相当于用很高的温度将关键的分布信息从原有的数据中分离，之后在同样的温度下用新模型融合蒸馏出来的数据分布，最后恢复温度，让两者充分融合。这也可以看成 Prof. Hinton 将这一个迁移学习过程命名为蒸馏的原因。



2.4 数学原理及验证



$$q_i = \frac{\exp(z_i/T)}{\sum_j \exp(z_j/T)}$$

z_i : the logit, i.e. the input to the softmax layer

q_i : the class probability computed by the softmax layer

T : a temperature that is normally set to 1

2.4.1 Loss Function

- **蒸馏类型：**在 softmax 层进行特征匹配
- **Soft target：**就是对应的带有 T 的目标，是要尽量的接近于大网络加入 T 后的分布概率。
- **HardTarget：**就是正常网络训练的目标，是要尽量的完成正确的分类。

两个目标函数分别为 soft target 和 hard target。在 Student Network 会有两个 loss，分别对应上面两个问题求得的交叉熵，作为小网络训练的 loss function。

2.4.2 具体蒸馏是如何训练的？

- **Teacher：**

  对 softmax（T=20）的输出与原始 label 求 loss。

- **Student：**

  (1)对 softmax（T=20）的输出与 Teacher 的 softmax（T=20）的输出求 loss1。

  (2)对 softmax（T=1）的输出与原始 label 求 loss2。

  (3)loss = loss1+loss2

2.4.3 实现方式（分两阶段）：

(1)原始模型训练阶段：

根据提出的目标问题，设计一个或多个复杂网络（N1，N2,…,Nt）。

收集足够的训练数据，按照常规 CNN 模型训练流程，并行的训练 1 中的多个网络得到。得到（M1,M2,…,Mt）

(2)浅层模型训练阶段：

根据（N1，N2,…,Nt）设计一个简单网络 N0。

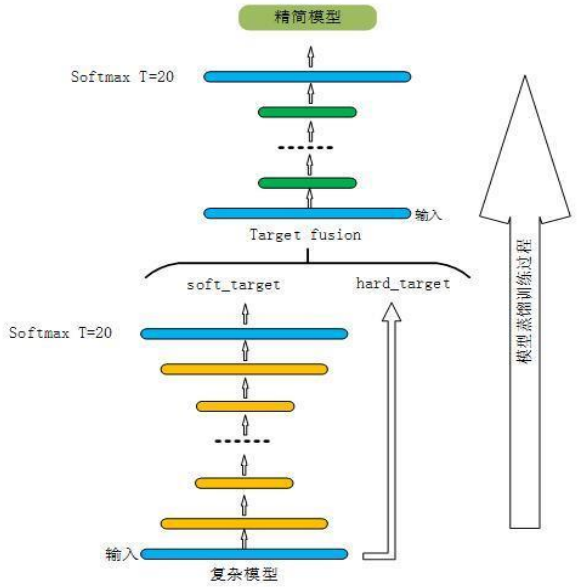
收集简单模型训练数据，此处的训练数据可以是训练原始网络的有标签数据，也可以是额外的无标签数据。

将 2 中收集到的样本输入原始模型（M1,M2,…,Mt），修改原始模型 softmax 层中温度参数 T 为一个较大值如 T=20。每一个样本在每个原始模型可以得到其最终的分类概率向量，选取其中概率至最大即为该模型对于当前样本的判定结果。对于 t 个原始模型就可以 t 概率

向量。然后对 t 概率向量求取均值作为当前样本最后的概率输出向量，记为 soft_target，保存。

标签融合 2 中收集到的数据定义为 hard_target，有标签数据的 hard_target 取值为其标签值 1，无标签数据 hard_taret 取值为 0。Target = ahard_target + bsoft_target（a+b=1）。Target 最终作为训练数据的标签去训练精简模型。参数 a，b 是用于控制标签融合权重的，推荐经验值为（a=0.1 b=0.9）

设置精简模型 softmax 层温度参数与原始复杂模型产生 Soft-target 时所采用的温度，按照常规模型训练精简网络模型。部署时将精简模型中的 softmax 温度参数重置为 1，即采用最原始的 softmax



Preliminary experiments on MNIST

A Single Large Neural Net:

- 2 hidden layers, 1200 rectified linear hidden units per layer, on all $60,000$ training cases, strongly regularized using dropout and weight-constraints[9] $\rightarrow$ 67 test errors

A Small Model:

- 2 hidden layers, 800 hidden units per layer, no regularization $\rightarrow$ 146 test errors
- additionally matching the soft targets of the large net at $T = 20$ $\rightarrow$ 74 test errors
  - 300 or more units per hidden layer: $T > 8$, fairly similar results
  - 30 units per hidden layer: $T \in [2.5, 4]$, significantly better than higher or lower temperatures
- omitting all examples of the digit 3 from the transfer set $\rightarrow$ 206 test errors ($133/1010$ threes)
  - fine-tune bias for the 3 class $\rightarrow$ 109 test errors ($14/1010$ threes)
- in the transfer set only containing the digit 7 and 8 from the training set: 47.3% test errors
  - fine-tune bias for the 7 and 8 class:13.2% test errors

将迁移数据集中的 3 或者 7、8 去掉是为了证明**小模型也能够从 soft target 中学得知识**。

Training ensembles of specialists on very big dataset

Training an ensemble of models:

- An ensemble requires too much computation at test time can be dealt with by using distillation.
- If the individual models are large neural networks and the dataset is very large, the amount of computation required at training time is excessive, even though it is easy to parallelize.

Learning specialist models:

- to show how learning specialist models that each focus on a different confusable subset of the classes can reduce the total amount of computation required to learn an ensemble
- to show how the overfitting of training specialist models may be prevented by using soft targets

2.5 开源实现(第三方)

复现性：无官方开源，下面使第三方实现：

- caffe 实现（作者只实现 cpu 版本，据说并不是特别拖速度）:knowledge_distillation_caffe
https://github.com/wentianli/knowledge_distillation_caffe
- keras 实现：knowledge-distillation-keras
https://github.com/TropComplique/knowledge-distillation-keras
- Tensorflow 实现：model_compression
https://github.com/chengshengchan/model_compression

注意事项：训练时将浅层网络温度与深层网络一致，部署时将浅层网络温度置 1.

3.研究进展

在 Knowledge Distillation 中有两个关键问题：

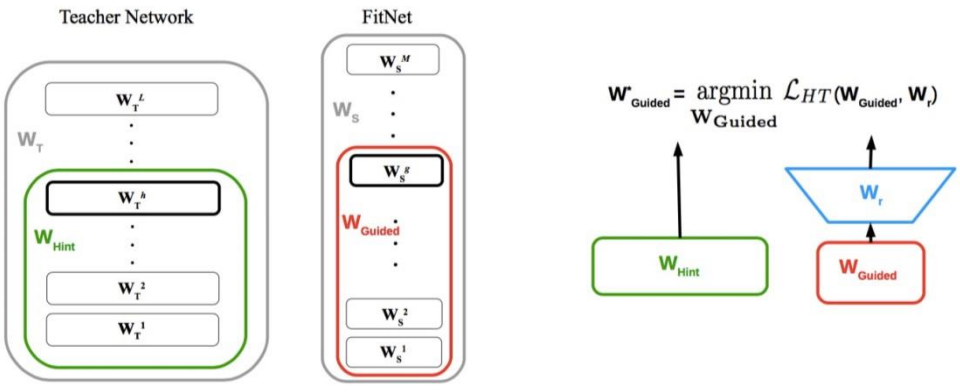
(1)如何定义知识;

(2)使用什么损失函数来度量 student 网络和 teacher 网络间相似度。

以下为 KD 方向的一些重要论文总结。

- **Hints for Thin Deep Nets, Romero， Bengio [4]**

在 Hinton 对 softmax 改造的基础上，对于中间层的权值匹配模拟学习，考虑到了 tea net 和 stu net 的 hidden layers features 的 map。首先说两边中间层选取，作者是这样解释的：The deeper we set the guided layer, the less flexibility we give to the network and, therefore, FitNets are more likely to suffer from over-regularization. 即太深了会容易过正则化，以及灵活性差。故都选了最中间的层。会存在的一个小问题是两个层的 dimension 不同，解决方案是通过一个 regressor 实现对齐，regressor 的选择上还有一些为了减少计算量进行的调整。



- **Attention Transfer, ICLR2017 [5]**

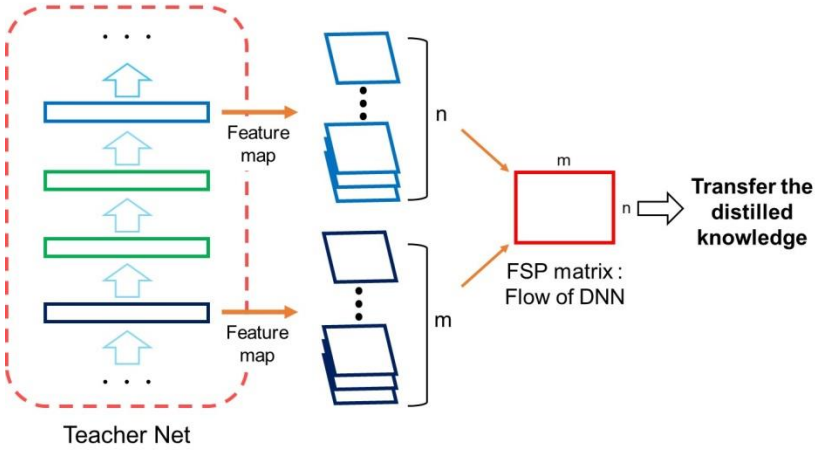
该文章启发于 CNN 中注意力机制，利用 teacher 模型中间层生成空间注意力图（spatial-attention），是一种热力图，图中越感兴趣地区颜色越红。作者借鉴 Distilling 的思想，使用复杂网络中能够提供视觉相关位置信息的 Attention map 来监督小网络的学习，并且结合了低、中、高三个层次的特征。在 feature map 中定义了 attention，使用了三种不同定义方法，将 attention 作为知识 transfer 到 student network 中。

Attention Transfer ，传递 teacher 网络的 attention 信息给 student 网络。首先，CNN 的

attention 一般分为两种，spatial-attention,channel-attention。本文利用的是 spatial-attention. 所谓 spatial-attention 即一种热力图，用来解码出输入图像空间区域对输出贡献大小。文章提出了两种可利用的 spatial-attention,基于响应图的和基于梯度图的。

- **FSP matrix, 2017 [6]**

它实际上将原始网络中 feature map 之间的相关度作为知识 transfer 到 student network 中，同时使用了 L2 损失函数。将 teacger 网络层与层之间的关系作为 student 网络 mimic 的目标。这篇文章介绍的这种知识蒸馏的方法类似风格迁移的 gram 矩阵。



- **DarkRank: Accelerating Deep Metric Learning via Cross Sample Similarities[7]**

该论文从一个新的视角对 Teacher 和 Student 网络间的 loss 进行了设计，将不同样本之间的相似性排序融入到监督训练中，并融合了 softmax、Verify loss、triplet loss 共同训练 student。(该方法在小数据集下的行人再识别上做的实验)

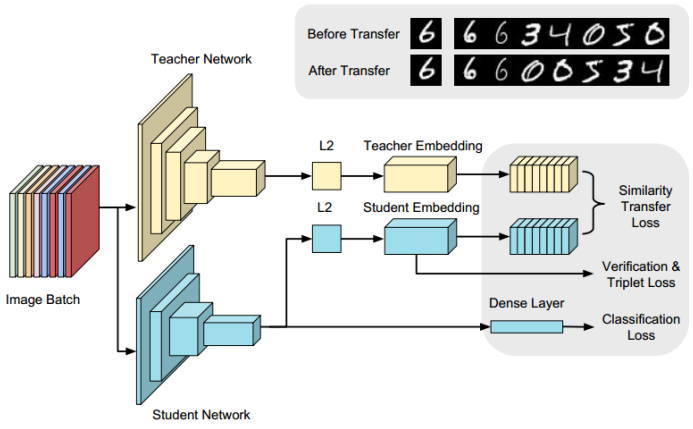


Figure 1: The network architecture of our DarkRank method.

参考文献:

[1] Hinton,Geoffrey, Oriol Vinyals, and Jeff Dean. "Distilling the knowledge in aneural network." arXiv preprint arXiv:1503.02531 (2015).

[2] Pan,Sinno Jialin, and Qiang Yang. "A survey on transfer learning." IEEE Transactionson knowledge and data engineering 22.10 (2010): 1345-1359.

[3] Buciluǎ, Cristian, Rich Caruana, andAlexandru Niculescu-Mizil. "Model compression." Proceedings of the12th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, 2006.

[4] Hints for Thin Deep Nets, Romero，Bengio

[5] Paying More Attention to Attention: Improving the Performance of Convolutional Neural Networks via Attention Transfer：https://arxiv.org/abs/1612.03928

[6] A Gift from Knowledge Distillation:Fast Optimization, Network Minimization and Transfer Learning，CVPR，2017

[7] DarkRank: Accelerating Deep Metric Learning via Cross Sample Similarities Transfer，2018，AAAI

https://www.jianshu.com/p/4893122112fa

https://blog.csdn.net/lijjianqing/article/details/79625041

https://blog.csdn.net/zhongshaoyy/article/details/53582048

https://blog.csdn.net/paper_reader/article/details/81080857

https://zhuanlan.zhihu.com/p/51563760 (知识蒸馏最新进展)

https://blog.csdn.net/shi2xian2wei2/article/details/84570620 (KD pytorch 实现及分析)

https://github.com/PolarisShi/distillation (知识蒸馏训练实例练手)

https://blog.csdn.net/qzrdypbuqk/article/details/81482598 (KD 总结全面、质量高)