Distilling the Knowledge in a Neural Network
蒸馏神经网络中的知识

Geoffrey Hinton 杰弗里·辛顿

Google Inc. Mountain View geoffhinton@google.com

谷歌公司 山景城 geoffhinton@google.com

&Oriol Vinyals[†] 奥里奥尔·维尼亚尔斯 [†]

Google Inc. Mountain View vinyals@google.com

谷歌公司 山景城 vinyals@google.com

&Jeff Dean Google Inc. Mountain View jeff@google.com

&杰夫·迪恩 谷歌公司 山景城 jeff@google.com

Also affiliated with the University of Toronto and the Canadian Institute for Advanced Research.Equal contribution.

同时隶属于多伦多大学和加拿大高等研究院。同等贡献。

Abstract 摘要

A very simple way to improve the performance of almost any machine learning algorithm is to train many different models on the same data and then to average their predictions [3]. Unfortunately, making predictions using a whole ensemble of models is cumbersome and may be too computationally expensive to allow deployment to a large number of users, especially if the individual models are large neural nets. Caruana and his collaborators [1] have shown that it is possible to compress the knowledge in an ensemble into a single model which is much easier to deploy and we develop this approach further using a different compression technique. We achieve some surprising results on MNIST and we show that we can significantly improve the acoustic model of a heavily used commercial system by distilling the knowledge in an ensemble of models into a single model. We also introduce a new type of ensemble composed of one or more full models and many specialist models which learn to distinguish fine-grained classes that the full models confuse. Unlike a mixture of experts, these specialist models can be trained rapidly and in parallel.

提升几乎所有机器学习算法性能的一个非常简单的方法是：在同一数据上训练多个不同模型，然后对其预测结果进行平均[3]。然而，使用整个模型集成进行预测不仅操作繁琐，且计算成本高昂，难以向大规模用户部署——尤其当单个模型是大型神经网络时。Caruana 及其合作者[1]已证明，将集成模型中的知识压缩至单一模型是可行的，这种模型更易于部署。我们采用不同的压缩技术进一步发展了该方法。我们在 MNIST 数据集上取得了令人惊喜的成果，并证明通过将集成模型的知识提炼到单一模型中，能显著改进某商用系统的声学模型性能。我们还提出了一种新型集成架构，该架构由一个或多个完整模型与若干专用模型组成，这些专用模型专门学习区分完整模型容易混淆的细粒度类别。与专家混合模型不同，这些专用模型可实现快速并行训练。

1 Introduction 1 引言

Many insects have a larval form that is optimized for extracting energy and nutrients from the environment and a completely different adult form that is optimized for the very different requirements of traveling and reproduction. In large-scale machine learning, we typically use very similar models for the training stage and the deployment stage despite their very different requirements: For tasks like speech and object recognition, training must extract structure from very large, highly redundant datasets but it does not need to operate in real time and it can use a huge amount of computation. Deployment to a large number of users, however, has much more stringent requirements on latency and computational resources. The analogy with insects suggests that we should be willing to train very cumbersome models if that makes it easier to extract structure from the data. The cumbersome model could be an ensemble of separately trained models or a single very large model trained with a very strong regularizer such as dropout [9]. Once the cumbersome model has been trained, we can then use a different kind of training, which we call “distillation” to transfer the knowledge from the cumbersome model to a small model that is more suitable for deployment. A version of this strategy has already been pioneered by Rich Caruana and his collaborators [1]. In their important paper they demonstrate convincingly that the knowledge acquired by a large ensemble of models can be transferred to a single small model.

许多昆虫拥有幼虫形态，其构造专为从环境中汲取能量与营养而优化；而成虫形态则截然不同，专为满足迁徙与繁殖的差异化需求而进化。在大规模机器学习中，尽管训练阶段与部署阶段的需求差异显著，我们却通常使用极为相似的模型：对于语音识别和物体识别等任务，训练过程需要从海量且高度冗余的数据集中提取结构特征，虽无需实时运行且可消耗巨大计算资源；然而面向海量用户的模型部署，则对延迟和计算资源有着更为严苛的要求。以昆虫为喻，启示我们应当勇于训练极其笨重的模型——只要这能更有效地从数据中提取结构信息。这种笨重模型可以是分别训练的模型集成，也可以是使用强正则化方法（如 Dropout[9]）单独训练的巨型模型。 一旦繁琐模型训练完成，我们便可采用另一种称为"蒸馏"的训练方式，将知识从繁琐模型迁移至更适合部署的小型模型中。这一策略的雏形早已由 Rich Caruana 及其合作者率先提出[1]。在他们具有重要意义的论文中，作者令人信服地证明了从大型模型集成中获取的知识能够有效迁移至单个小型模型。

A conceptual block that may have prevented more investigation of this very promising approach is that we tend to identify the knowledge in a trained model with the learned parameter values and this makes it hard to see how we can change the form of the model but keep the same knowledge. A more abstract view of the knowledge, that frees it from any particular instantiation, is that it is a learned mapping from input vectors to output vectors. For cumbersome models that learn to discriminate between a large number of classes, the normal training objective is to maximize the average log probability of the correct answer, but a side-effect of the learning is that the trained model assigns probabilities to all of the incorrect answers and even when these probabilities are very small, some of them are much larger than others. The relative probabilities of incorrect answers tell us a lot about how the cumbersome model tends to generalize. An image of a BMW, for example, may only have a very small chance of being mistaken for a garbage truck, but that mistake is still many times more probable than mistaking it for a carrot.

一个可能阻碍了对这一极具前景方法进行更多研究的概念障碍是，我们往往将训练后模型中的知识与学习到的参数值等同起来，这使得我们难以理解如何在改变模型形式的同时保留相同的知识。对知识更为抽象的理解是将其视为从输入向量到输出向量的学习映射，这种理解使其摆脱了任何具体实例的束缚。对于学习区分大量类别的复杂模型而言，常规训练目标是最大化正确答案的平均对数概率，但学习的一个副作用是训练后的模型会为所有错误答案分配概率，即使这些概率非常小，其中某些错误答案的概率仍远高于其他。错误答案的相对概率揭示了复杂模型泛化的重要规律。例如，一张宝马汽车的照片被误认为垃圾车的概率可能微乎其微，但这一错误概率仍远高于将其误认为胡萝卜的概率。

It is generally accepted that the objective function used for training should reflect the true objective of the user as closely as possible. Despite this, models are usually trained to optimize performance on the training data when the real objective is to generalize well to new data. It would clearly be better to train models to generalize well, but this requires information about the correct way to generalize and this information is not normally available. When we are distilling the knowledge from a large model into a small one, however, we can train the small model to generalize in the same way as the large model. If the cumbersome model generalizes well because, for example, it is the average of a large ensemble of different models, a small model trained to generalize in the same way will typically do much better on test data than a small model that is trained in the normal way on the same training set as was used to train the ensemble.

普遍认为，训练所用的目标函数应尽可能贴近用户的真实目标。尽管如此，模型通常被训练以优化在训练数据上的表现，而真正的目标却是对新数据具有良好的泛化能力。显然，训练模型以实现良好泛化更为理想，但这需要关于正确泛化方式的信息，而这类信息通常难以获取。然而，当我们将知识从大型模型提炼到小型模型时，我们可以训练小型模型以与大型模型相同的方式进行泛化。如果复杂模型（例如，作为多个不同模型集成平均的结果）能够良好泛化，那么以相同方式训练的小型模型在测试数据上的表现通常会远优于使用常规方法、在与训练集成相同的数据集上训练的小型模型。

An obvious way to transfer the generalization ability of the cumbersome model to a small model is to use the class probabilities produced by the cumbersome model as “soft targets” for training the small model. For this transfer stage, we could use the same training set or a separate “transfer” set. When the cumbersome model is a large ensemble of simpler models, we can use an arithmetic or geometric mean of their individual predictive distributions as the soft targets. When the soft targets have high entropy, they provide much more information per training case than hard targets and much less variance in the gradient between training cases, so the small model can often be trained on much less data than the original cumbersome model and using a much higher learning rate.

将笨重模型的泛化能力迁移至小型模型的一个明显方法是，使用笨重模型产生的类别概率作为训练小型模型的“软目标”。在此迁移阶段，我们可以使用相同的训练集或单独的“迁移”数据集。当笨重模型是由多个简单模型组成的大型集成时，我们可以使用它们各自预测分布的算术平均或几何平均作为软目标。当软目标具有高熵时，每个训练样本所提供的信息量远大于硬目标，且训练样本间梯度的方差显著降低，因此小型模型通常可以用比原笨重模型少得多的数据量进行训练，并采用更高的学习率。

For tasks like MNIST in which the cumbersome model almost always produces the correct answer with very high confidence, much of the information about the learned function resides in the ratios of very small probabilities in the soft targets. For example, one version of a 2 may be given a probability of 10^{-6} of being a 3 and 10^{-9} of being a 7 whereas for another version it may be the other way around. This is valuable information that defines a rich similarity structure over the data (*i. e.* it says which 2's look like 3's and which look like 7's) but it has very little influence on the cross-entropy cost function during the transfer stage because the probabilities are so close to zero. Caruana and his collaborators circumvent this problem by using the logits (the inputs to the final softmax) rather than the probabilities produced by the softmax as the targets for learning the small model and they minimize the squared difference between the logits produced by the cumbersome model and the logits produced by the small model. Our more general solution, called "distillation", is to raise the temperature of the final softmax until the cumbersome model produces a suitably soft set of targets. We then use the same high temperature when training the small model to match these soft targets. We show later that matching the logits of the cumbersome model is actually a special case of distillation.

对于像 MNIST 这样的任务，笨重模型几乎总是以极高的置信度给出正确答案，此时关于已学习函数的大量信息蕴含在软目标中极小的概率比值里。例如，某个数字“2”的变体可能被赋予成为“3”的概率为 10^{-6} ，成为“7”的概率为 10^{-9} ，而另一个变体的情况可能恰恰相反。这些宝贵信息定义了数据间丰富的相似性结构（即指明哪些“2”看起来像“3”，哪些像“7”），但在迁移阶段对交叉熵损失函数的影响微乎其微，因为这些概率值无限趋近于零。Caruana 及其合作者通过采用 logits（最终 softmax 层的输入）而非 softmax 输出的概率作为小模型的学习目标，成功规避了这一问题——他们通过最小化笨重模型与小模型产生的 logits 之间的平方差来实现这一目标。 我们提出的更通用的解决方案称为“蒸馏”，即提高最终 softmax 的温度，直到复杂模型生成足够柔和的目标分布。随后在训练小型模型时使用相同的高温参数，以匹配这些柔和的目标。后文将证明，匹配复杂模型的 logits 实际上是蒸馏的一种特例。

The transfer set that is used to train the small model could consist entirely of unlabeled data [1] or we could use the original training set. We have found that using the original training set works well, especially if we add a small term to the objective function that encourages the small model to predict the true targets as well as matching the soft targets provided by the cumbersome model. Typically, the small model cannot exactly match the soft targets and erring in the direction of the correct answer turns out to be helpful.

用于训练小型模型的迁移数据集可以完全由未标注数据构成[1]，也可以使用原始训练集。我们发现使用原始训练集效果显著，特别是在目标函数中加入小型项，既促使小型模型预测真实目标，又要求其匹配复杂模型提供的柔和目标。通常情况下，小型模型无法完全匹配柔和目标，而向正确答案方向的适度偏差反而能提升模型性能。

2 Distillation 2 蒸馏

Neural networks typically produce class probabilities by using a "softmax" output layer that converts the logit, z_i, computed for each class into a probability, q_i, by comparing z_i with the other logits.

神经网络通常通过使用“softmax”输出层来生成类别概率，该层将每个类别计算出的逻辑值 z_i 转换为概率 q_i ，其方法是将 z_i 与其他逻辑值进行比较。

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

where T is a temperature that is normally set to 1. Using a higher value for T produces a softer probability distribution over classes.

其中 T 是温度参数，通常设置为 1 。使用较高的 T 值会在类别上产生更柔和的概率分布。

In the simplest form of distillation, knowledge is transferred to the distilled model by training it on a transfer set and using a soft target distribution for each case in the transfer set that is produced by using the cumbersome model with a high temperature in its softmax. The same high temperature is used when training the distilled model, but after it has been trained it uses a temperature of 1.

在最简单的蒸馏形式中，知识通过以下方式传递到蒸馏模型：在迁移集上训练模型，并对迁移集中的每个样本使用由复杂模型在其 softmax 中使用高温产生的软目标分布。训练蒸馏模型时使用相同的高温参数，但在训练完成后会将温度设置为 1。

When the correct labels are known for all or some of the transfer set, this method can be significantly improved by also training the distilled model to produce the correct labels. One way to do this is to use the correct labels to modify the soft targets, but we found that a better way is to simply use a weighted average of two different objective functions. The first objective function is the cross entropy with the soft targets and this cross entropy is computed using the same high temperature in the softmax of the distilled model as was used for generating the soft targets from the cumbersome model. The second objective function is the cross entropy with the correct labels. This is computed using exactly the same logits in softmax of the distilled model but at a temperature of 1. We found that the best results were generally obtained by using a condiderably lower weight on the second objective function. Since the magnitudes of the gradients produced by the soft targets scale as $1/T^2$ it is important to multiply them by T^2 when using both hard and soft targets. This ensures that the relative contributions of the hard and soft targets remain roughly unchanged if the temperature used for distillation is changed while experimenting with meta-parameters.

当迁移集中的所有或部分数据已知正确标签时，通过同时训练蒸馏模型生成正确标签可显著提升该方法效果。一种改进方式是利用正确标签修正软目标，但我们发现更优方案是直接采用两个目标函数的加权平均值。第一个目标函数是基于软目标的交叉熵，计算时蒸馏模型的 softmax 使用与笨重模型生成软目标时相同的高温度参数。第二个目标函数是基于真实标签的交叉熵，计算时使用蒸馏模型 softmax 的相同逻辑输出值，但温度参数设为 1。我们发现最佳结果通常是通过对第二个目标函数赋予相对较低的权重实现的。由于软目标产生的梯度量级约为 $1/T^2$ ，当同时使用硬目标和软目标时，必须将其乘以 T^2 进行缩放。 这确保了在实验元参数时，若改变用于蒸馏的温度，硬目标和软目标的相对贡献大致保持不变。

2.1 Matching logits is a special case of distillation

2.1 匹配逻辑值是蒸馏的一种特殊情况

Each case in the transfer set contributes a cross-entropy gradient, dC/dz_i, with respect to each logit, z_i of the distilled model. If the cumbersome model has logits v_i which produce soft target probabilities p_i and the transfer training is done at a temperature of T, this gradient is given by:

迁移集中的每个样本都会对蒸馏模型的每个逻辑值 z_i 产生交叉熵梯度 dC/dz_i 。若复杂模型具有产生软目标概率 p_i 的逻辑值 v_i ，且迁移训练在温度 T 下进行，则该梯度可表示为：

$$\frac{\partial C}{\partial z_i} = \frac{1}{T}\left(q_i - p_i\right) = \frac{1}{T}\left(\frac{e^{z_i/T}}{\sum_j e^{z_j/T}} - \frac{e^{v_i/T}}{\sum_j e^{v_j/T}}\right) \tag{2}$$

If the temperature is high compared with the magnitude of the logits, we can approximate:

若温度远高于逻辑值的量级，我们可以近似得出：

$$\frac{\partial C}{\partial z_i} \approx \frac{1}{T}\left(\frac{1 + z_i/T}{N + \sum_j z_j/T} - \frac{1 + v_i/T}{N + \sum_j v_j/T}\right) \tag{3}$$

If we now assume that the logits have been zero-meaned separately for each transfer case so that $\sum_j z_j = \sum_j v_j = 0$ Eq. 3 simplifies to:

如果我们现在假设对于每个迁移案例，logits 已分别进行零均值处理，那么 $\sum_j z_j = \sum_j v_j = 0$ 公式 3 可简化为：

$$\frac{\partial C}{\partial z_i} \approx \frac{1}{NT^2}\left(z_i - v_i\right) \tag{4}$$

So in the high temperature limit, distillation is equivalent to minimizing $1/2(z_i - v_i)^2$, provided the logits are zero-meaned separately for each transfer case. At lower temperatures, distillation pays much less attention to matching logits that are much more negative than the average. This is potentially advantageous because these logits are almost completely unconstrained by the cost function used for training the cumbersome model so they could be very noisy. On the other hand, the very negative logits may convey useful information about the knowledge acquired by the cumbersome model. Which of these effects dominates is an empirical question. We show that when the distilled model is much too small to capture all of the knowledege in the cumbersome model, intermediate temperatures work best which strongly suggests that ignoring the large negative logits can be helpful.

因此在高温极限下，只要对每个迁移案例分别进行 logits 零均值处理，蒸馏就等价于最小化 $1/2(z_i - v_i)^2$ 。在较低温度下，蒸馏会大幅减少对那些远低于平均值的负 logits 的匹配关注。这具有潜在优势，因为这些 logits 几乎完全不受笨重模型训练所用成本函数的约束，可能包含大量噪声。另一方面，这些极端负值的 logits 可能传递着笨重模型所获知识的重要信息。这两种效应孰主孰次属于实证研究问题。我们的实验表明，当蒸馏模型规模过小而无法完全捕捉笨重模型中的所有知识时，中等温度效果最佳，这充分证明忽略大幅负值的 logits 可能带来益处。

3 Preliminary experiments on MNIST

在 MNIST 数据集上的初步实验

To see how well distillation works, we trained a single large neural net with two hidden layers of 1200 rectified linear hidden units on all 60,000 training cases. The net was strongly regularized using dropout and weight-constraints as described in [5]. Dropout can be viewed as a way of training an exponentially large ensemble of models that share weights. In addition, the input images were jittered by up to two pixels in any direction. This net achieved 67 test errors whereas a smaller net with two hidden layers of 800 rectified linear hidden units and no regularization achieved 146 errors. But if the smaller net was regularized solely by adding the additional task of matching the soft targets produced by the large net at a temperature of 20, it achieved 74 test errors. This shows that soft targets can transfer a great deal of knowledge to the distilled model, including the knowledge about how to generalize that is learned from translated training data even though the transfer set does not contain any translations.

为验证蒸馏效果，我们在全部 6 万组训练样本上训练了一个大型神经网络，该网络包含两个隐藏层，每层配备 1200 个修正线性单元。该网络采用文献[5]所述的 dropout 与权重约束方法进行强正则化处理。Dropout 可视为训练共享权重的指数级规模模型集成的一种方式。此外，输入图像在任意方向上进行最多两个像素的随机扰动。该网络实现了 67 个测试误差，而仅配备 800 个修正线性隐藏单元且未进行正则化的小型网络则产生 146 个误差。但若通过额外添加匹配大型网络在温度参数 20 下生成的软目标这一任务对小型网络进行正则化，其测试误差降至 74 个。这表明软目标能够向蒸馏模型传递大量知识，包括从平移训练数据中学到的泛化能力——尽管迁移数据集本身并未包含任何平移样本。

When the distilled net had 300 or more units in each of its two hidden layers, all temperatures above 8 gave fairly similar results. But when this was radically reduced to 30 units per layer, temperatures in the range 2.5 to 4 worked significantly better than higher or lower temperatures.

当蒸馏网络的两个隐藏层各有 300 个或更多单元时，所有高于 8 的温度值都给出了相当接近的结果。但当每层单元数大幅减少至 30 个时，2.5 到 4 范围内的温度值表现明显优于更高或更低的温度值。

We then tried omitting all examples of the digit 3 from the transfer set. So from the perspective of the distilled model, 3 is a mythical digit that it has never seen. Despite this, the distilled model only makes 206 test errors of which 133 are on the 1010 threes in the test set. Most of the errors are caused by the fact that the learned bias for the 3 class is much too low. If this bias is increased by 3.5 (which optimizes overall performance on the test set), the distilled model makes 109 errors of which 14 are on 3s. So with the right bias, the distilled model gets 98.6% of the test 3s correct despite never having seen a 3 during training. If the transfer set contains *only* the 7s and 8s from the training set, the distilled model makes 47.3% test errors, but when the biases for 7 and 8 are reduced by 7.6 to optimize test performance, this falls to 13.2% test errors.

随后，我们尝试从迁移集中剔除所有数字 3 的样本。因此，从蒸馏模型的角度来看，3 是一个它从未见过的神秘数字。尽管如此，蒸馏模型仅产生 206 个测试错误，其中 133 个出现在测试集的 1010 个数字 3 上。大部分错误源于学习到的 3 类偏置值过低。若将该偏置增加 3.5（此操作可优化测试集整体性能），蒸馏模型的错误数降至 109 个，其中仅 14 个涉及数字 3。通过调整偏置，蒸馏模型对测试集中数字 3 的识别准确率高达 98.6%，尽管其在训练过程中从未接触过数字 3。当迁移集仅包含训练集中的数字 7 和 8 时，蒸馏模型的测试错误率达 47.3%，但将 7 和 8 的偏置降低 7.6 以优化测试性能后，错误率骤降至 13.2%。

4 Experiments on speech recognition

4 语音识别实验

In this section, we investigate the effects of ensembling Deep Neural Network (DNN) acoustic models that are used in Automatic Speech Recognition (ASR). We show that the distillation strategy that we propose in this paper achieves the desired effect of distilling an ensemble of models into a single model that works significantly better than a model of the same size that is learned directly from the same training data.

在本节中，我们研究了集成深度神经网络声学模型在自动语音识别系统中的效果。我们证明，本文提出的知识蒸馏策略成功实现了将模型集合提炼为单一模型的目标，该单一模型的表现显著优于直接从相同训练数据学习的同规模模型。

State-of-the-art ASR systems currently use DNNs to map a (short) temporal context of features derived from the waveform to a probability distribution over the discrete states of a Hidden Markov Model (HMM) [4]. More specifically, the DNN produces a probability distribution over clusters of tri-phone states at each time and a decoder then finds a path through the HMM states that is the best compromise between using high probability states and producing a transcription that is probable under the language model.

当前最先进的自动语音识别系统采用深度神经网络，将从波形中提取的（短时）时序特征映射至隐马尔可夫模型离散状态的概率分布[4]。具体而言，深度神经网络在每个时间步生成三音素状态簇的概率分布，随后解码器通过寻找在隐马尔可夫模型状态空间中的最优路径，实现高概率状态使用与语言模型下转录文本概率之间的最佳平衡。

Although it is possible (and desirable) to train the DNN in such a way that the decoder (and, thus, the language model) is taken into account by marginalizing over all possible paths, it is common to train the DNN to perform frame-by-frame classification by (locally) minimizing the cross entropy between the predictions made by the net and the labels given by a forced alignment with the ground truth sequence of states for each observation:

尽管可以（且理想情况下应该）通过边缘化所有可能路径来训练深度神经网络（DNN），以考虑解码器（及语言模型）的影响，但通常的做法是通过（局部）最小化网络预测与通过强制对齐得到的每个观测对应的真实状态序列标签之间的交叉熵，来训练 DNN 进行逐帧分类：

$$\boldsymbol{\theta} = \arg\max_{\boldsymbol{\theta}'} P(h_t \mid \mathbf{s}_t; \boldsymbol{\theta}')$$

where $\boldsymbol{\theta}$ are the parameters of our acoustic model P which maps acoustic observations at time t, $\mathbf{s}_t$, to a probability, $P(h_t \mid \mathbf{s}_t; \boldsymbol{\theta}')$, of the “correct” HMM state h_t, which is determined by a forced alignment with the correct sequence of words. The model is trained with a distributed stochastic gradient descent approach.

其中 $\boldsymbol{\theta}$ 是我们声学模型 P 的参数，该模型将时间 t 的声学观测 $\mathbf{s}_t$ 映射到“正确”隐马尔可夫模型状态 h_t 的概率 $P(h_t \mid \mathbf{s}_t; \boldsymbol{\theta}')$ ，该状态通过与正确词序列的强制对齐确定。模型采用分布式随机梯度下降方法进行训练。

We use an architecture with 8 hidden layers each containing 2560 rectified linear units and a final softmax layer with 14,000 labels (HMM targets h_t). The input is 26 frames of 40 Mel-scaled filterbank coefficients with a 10ms advance per frame and we predict the HMM state of 21st frame. The total number of parameters is about 85M. This is a slightly outdated version of the acoustic model used by Android voice search, and should be considered as a very strong baseline. To train the DNN acoustic model we use about 2000 hours of spoken English data, which yields about 700M training examples. This system achieves a frame accuracy of 58.9%, and a Word Error Rate (WER) of 10.9% on our development set.

我们采用包含 8 个隐藏层的架构，每层具有 2560 个修正线性单元，以及一个包含 14,000 个标签（对应 HMM 目标 h_t ）的最终 softmax 层。输入为 26 帧 40 维梅尔尺度滤波器组系数，每帧时间步进为 10 毫秒，我们预测的是第 21 帧 st 的 HMM 状态。模型参数总量约为 8500 万。这是安卓语音搜索所用声学模型的稍旧版本，应被视为非常强大的基线系统。为训练该 DNN 声学模型，我们使用了约 2000 小时的英语语音数据，生成约 7 亿个训练样本。该系统在我们的开发集上实现了 58.9%的帧准确率，以及 10.9%的词错误率（WER）。

4.1 Results 4.1 结果

We trained 10 separate models to predict $P(h_t \mid \mathbf{s}_t; \boldsymbol{\theta})$, using exactly the same architecture and training procedure as the baseline. The models are randomly initialized with different initial parameter values and we find that this creates sufficient diversity in the trained models to allow the averaged predictions of the ensemble to significantly outperform the individual models. We have explored adding diversity to the models by varying the sets of data that each model sees, but we found this to not significantly change our results, so we opted for the simpler approach. For the distillation we tried temperatures of $[1, \mathbf{2}, 5, 10]$ and used a relative weight of 0.5 on the cross-entropy for the hard targets, where bold font indicates the best value that was used for table 1 .

我们训练了 10 个独立的模型来预测 $P(h_t \mid \mathbf{s}_t; \boldsymbol{\theta})$ ，采用与基线完全相同的架构和训练流程。这些模型通过不同的初始参数值随机初始化，我们发现这能在训练后的模型中产生足够的多样性，使得集成模型的平均预测表现显著优于单个模型。我们曾尝试通过让每个模型接触不同数据集来增加多样性，但发现这并未显著改变结果，因此选择了更简便的方法。在蒸馏过程中，我们尝试了 $[1, \mathbf{2}, 5, 10]$ 的温度值，并对硬目标的交叉熵损失采用了 0.5 的相对权重，其中加粗字体表示用于表 1 的最佳数值。

Table 1 shows that, indeed, our distillation approach is able to extract more useful information from the training set than simply using the hard labels to train a single model. More than 80% of the improvement in frame classification accuracy achieved by using an ensemble of 10 models is transferred to the distilled model which is similar to the improvement we observed in our preliminary experiments on MNIST. The ensemble gives a smaller improvement on the ultimate objective of WER (on a 23K-word test set) due to the mismatch in the objective function, but again, the improvement in WER achieved by the ensemble is transferred to the distilled model.

表 1 显示，我们的蒸馏方法确实能够从训练集中提取比仅使用硬标签训练单一模型更有用的信息。通过使用 10 个模型组成的集成所实现的帧分类准确率提升中，超过 80%被转移到了蒸馏模型中，这与我们在 MNIST 初步实验中观察到的改进程度相似。由于目标函数不匹配，集成模型在最终目标 WER（基于 23K 词测试集）上的改进较小，但集成模型实现的 WER 提升同样被完整转移到了蒸馏模型中。

System 系统	Test Frame Accuracy 测试框架准确率	WER
Baseline 基线	58.9%	10.9%
10xEnsemble 10 倍集成	61.1%	10.7%
Distilled Single model 蒸馏单一模型	60.8%	10.7%

Table 1: Frame classification accuracy and WER showing that the distilled single model performs about as well as the averaged predictions of 10 models that were used to create the soft targets.

表 1：帧分类准确率和词错误率显示，蒸馏后的单一模型表现与用于生成软目标的 10 个模型平均预测结果相当。

We have recently become aware of related work on learning a small acoustic model by matching the class probabilities of an already trained larger model [8]. However, they do the distillation at a temperature of 1 using a large unlabeled dataset and their best distilled model only reduces the error rate of the small model by 28% of the gap between the error rates of the large and small models when they are both trained with hard labels.

我们最近了解到通过匹配已训练大型模型的类别概率来学习小型声学模型的相关研究[8]。然而，他们在温度参数为 1 的情况下使用大型未标注数据集进行蒸馏，且其最佳蒸馏模型仅将小模型错误率降低了大型模型与小模型在硬标签训练下错误率差距的 28%。

5 Training ensembles of specialists on very big datasets

5 在超大规模数据集上训练专家集成模型

Training an ensemble of models is a very simple way to take advantage of parallel computation and the usual objection that an ensemble requires too much computation at test time can be dealt with by using distillation. There is, however, another important objection to ensembles: 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.

训练模型集成是利用并行计算的简单方法，而通常关于集成在测试阶段计算量过大的反对意见可以通过蒸馏技术来解决。然而，集成方法还存在另一个重要缺陷：若单个模型是大型神经网络且数据集规模极大，即使易于并行化，训练阶段所需的计算量仍会过于庞大。

In this section we give an example of such a dataset and we 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. The main problem with specialists that focus on making fine-grained distinctions is that they overfit very easily and we describe how this overfitting may be prevented by using soft targets.

在本节中，我们给出了此类数据集的一个示例，并展示了如何通过学习专注于不同易混淆类别子集的专家模型来减少学习集成模型所需的总计算量。专注于细粒度区分的专家模型主要问题在于它们极易过拟合，我们描述了如何通过使用软目标来防止这种过拟合现象。

5.1 The JFT dataset 5.1 JFT 数据集

JFT is an internal Google dataset that has 100 million labeled images with 15,000 labels. When we did this work, Google's baseline model for JFT was a deep convolutional neural network [7] that had been trained for about six months using asynchronous stochastic gradient descent on a large number of cores. This training used two types of parallelism [2]. First, there were many replicas of the neural net running on different sets of cores and processing different mini-batches from the training set. Each replica computes the average gradient on its current mini-batch and sends this gradient to a sharded parameter server which sends back new values for the parameters. These new values reflect all of the gradients received by the parameter server since the last time it sent parameters to the replica. Second, each replica is spread over multiple cores by putting different subsets of the neurons on each core. Ensemble training is yet a third type of parallelism that can be wrapped around the other two types, but only if a lot more cores are available. Waiting for several years to train an ensemble of models was not an option, so we needed a much faster way to improve the baseline model.

JFT 是谷歌内部数据集，包含 1 亿张标注图像及 1.5 万个分类标签。开展此项研究时，谷歌针对 JFT 的基准模型是基于深度卷积神经网络[7]构建的，该网络通过异步随机梯度下降算法在大量计算核心上进行了约六个月的训练。此训练过程采用两种并行化策略[2]：首先，在多组计算核心上运行神经网络的大量副本，每个副本处理训练集中不同的迷你批次数据。各副本计算当前迷你批次的平均梯度后，将梯度发送至分片参数服务器，参数服务器随即返回更新后的参数值——这些新参数值整合了自上次向该副本发送参数后服务器接收到的全部梯度。其次，通过将神经元的不同子集分配到不同计算核心，每个神经网络副本可跨多个核心并行运算。集成训练可作为第三种并行化方式与前两种结合运用，但这需要大量额外的计算核心。由于等待数年时间训练模型集成并不可行，我们亟需更高效的方法来提升基准模型性能。

5.2 Specialist Models 5.2 专家模型

When the number of classes is very large, it makes sense for the cumbersome model to be an ensemble that contains one generalist model trained on all the data and many “specialist” models, each of which is trained on data that is highly enriched in examples from a very confusable subset of the classes (like different types of mushroom). The softmax of this type of specialist can be made much smaller by combining all of the classes it does not care about into a single dustbin class.

当类别数量非常庞大时，采用集成式复杂模型是合理的方案——该集成模型包含一个基于全部数据训练的通用模型，以及多个“专家”模型。每个专家模型专门针对高度集中的易混淆类别子集（例如不同种类的蘑菇）进行训练。通过将此类专家模型不关注的类别统一归入单个“回收站”类别，可大幅缩减其 softmax 层的规模。

To reduce overfitting and share the work of learning lower level feature detectors, each specialist model is initialized with the weights of the generalist model. These weights are then slightly modified by training the specialist with half its examples coming from its special subset and half sampled at random from the remainder of the training set. After training, we can correct for the biased training set by incrementing the logit of the dustbin class by the log of the proportion by which the specialist class is oversampled.

为减少过拟合并共享底层特征检测器的学习成果，每个专家模型均采用通用模型的权重进行初始化。随后通过混合训练对权重进行微调：其中半数样本来自专家专属子集，另外半数从训练集剩余部分随机抽取。训练完成后，可通过按专家类别过采样比例的对数值增加回收站类别的 logit 值，从而修正训练集的偏差。

5.3 Assigning classes to specialists

5.3 为专家模型分配类别

In order to derive groupings of object categories for the specialists, we decided to focus on categories that our full network often confuses. Even though we could have computed the confusion matrix and used it as a way to find such clusters, we opted for a simpler approach that does not require the true labels to construct the clusters.

为了为专家模型推导出对象类别的分组，我们决定重点关注完整网络经常混淆的类别。尽管我们可以计算混淆矩阵并以此作为寻找此类聚类的方法，但我们选择了一种更简便的途径——无需真实标签即可构建这些聚类。

JFT 1: Tea party; Easter; Bridal shower; Baby shower; Easter Bunny; …

JFT 1：茶话会；复活节；新娘送礼会；宝宝送礼会；复活节兔子；……

JFT 2: Bridge; Cable-stayed bridge; Suspension bridge; Viaduct; Chimney; …

JFT 2：桥梁；斜拉桥；悬索桥；高架桥；烟囱；……

JFT 3: Toyota Corolla E100; Opel Signum; Opel Astra; Mazda Familia; …

JFT 3：丰田卡罗拉 E100；欧宝赛飞利；欧宝雅特；马自达 Familia；……

Table 2: Example classes from clusters computed by our covariance matrix clustering algorithm

表 2：通过我们的协方差矩阵聚类算法计算得出的聚类示例类别

In particular, we apply a clustering algorithm to the covariance matrix of the predictions of our generalist model, so that a set of classes S^m that are often predicted together will be used as targets for one of our specialist models, m. We applied an on-line version of the K-means algorithm to the columns of the covariance matrix, and obtained reasonable clusters (shown in Table 2). We tried several clustering algorithms which produced similar results.

具体而言，我们对通用模型预测结果的协方差矩阵应用聚类算法，使得一组经常被同时预测的类别 S^m 将作为我们某个专家模型 m 的学习目标。我们采用在线版 K 均值算法对协方差矩阵的列向量进行聚类，获得了合理的分类结果（如表 2 所示）。尝试的多种聚类算法均取得了相似效果。

5.4 Performing inference with ensembles of specialists

5.4 使用专家模型集成进行推理

Before investigating what happens when specialist models are distilled, we wanted to see how well ensembles containing specialists performed. In addition to the specialist models, we always have a generalist model so that we can deal with classes for which we have no specialists and so that we can decide which specialists to use. Given an input image $\mathbf{x}$, we do top-one classification in two steps:

在研究专家模型蒸馏效果之前，我们首先评估包含专家模型的集成系统性能。除专家模型外，我们始终保留通用模型以处理未分配专家模型的类别，并确定需要启用的专家模型。给定输入图像 $\mathbf{x}$ 后，我们通过两个步骤完成 top-1 分类：

Step 1: For each test case, we find the n most probable classes according to the generalist model. Call this set of classes k. In our experiments, we used $n=1$.

步骤 1：针对每个测试样本，根据通用模型找出 n 个最可能的类别，记作类别集合 k 。本实验中我们设定 $n=1$ 。

Step 2: We then take all the specialist models, m, whose special subset of confusable classes, S^m, has a non-empty intersection with k and call this the active set of specialists A_k (note that this set may be empty). We then find the full probability distribution $\mathbf{q}$ over all the classes that minimizes:

第二步：我们随后选取所有专家模型 m ，其特定的易混淆类别子集 S^m 与 k 存在非空交集，并将这些模型称为活跃专家集合 A_k （注意该集合可能为空）。接着我们求解覆盖所有类别的完整概率分布 $\mathbf{q}$ ，使其最小化以下目标：

$$KL(\mathbf{p}^g,\mathbf{q})+\sum_{m\,\in\,A_k}KL(\mathbf{p}^m,\mathbf{q}) \tag{5}$$

where KL denotes the KL divergence, and $\mathbf{p}^m$ $\mathbf{p}^g$ denote the probability distribution of a specialist model or the generalist full model. The distribution $\mathbf{p}^m$ is a distribution over all the specialist classes of m plus a single dustbin class, so when computing its KL divergence from the full $\mathbf{q}$ distribution we sum all of the probabilities that the full $\mathbf{q}$ distribution assigns to all the classes in m's dustbin.

其中 KL 表示 KL 散度， $\mathbf{p}^m$ $\mathbf{p}^g$ 分别代表专家模型或通用完整模型的概率分布。分布 $\mathbf{p}^m$ 是 m 所有专家类别加上单个回收类别的概率分布，因此在计算其与完整 $\mathbf{q}$ 分布的 KL 散度时，我们需要累加完整 $\mathbf{q}$ 分布赋予 m 回收类别中所有类别的概率值。

Eq. 5 does not have a general closed form solution, though when all the models produce a single probability for each class the solution is either the arithmetic or geometric mean, depending on whether we use $KL(\mathbf{p},\mathbf{q})$ or $KL(\mathbf{q},\mathbf{p})$). We parameterize $\mathbf{q}=softmax(\mathbf{z})$ (with $T=1$) and we use gradient descent to optimize the logits $\mathbf{z}$ w.r.t. eq. 5. Note that this optimization must be carried out for each image.

虽然公式 5 不存在通用闭式解，但当所有模型对每个类别仅生成单一概率时，解的形式取决于使用 $KL(\mathbf{p},\mathbf{q})$ 还是 $KL(\mathbf{q},\mathbf{p})$ ，分别对应算术平均或几何平均。我们通过参数化 $\mathbf{q}=softmax(\mathbf{z})$ （使用 $T=1$ ）并采用梯度下降法对逻辑值 $\mathbf{z}$ 进行优化以求解公式 5。需要注意的是，此项优化需针对每张图像单独执行。

5.5 Results 5.5 结果

System 系统	Conditional Test Accuracy 条件测试准确率	Test Accuracy 测试准确率
Baseline 基线	43.1%	25.0%
+ 61 Specialist models + 61 个专家模型	45.9%	26.1%

Table 3: Classification accuracy (top 1) on the JFT development set.

表 3：JFT 开发集上的分类准确率（前 1）

Starting from the trained baseline full network, the specialists train extremely fast (a few days instead of many weeks for JFT). Also, all the specialists are trained completely independently. Table 3 shows the absolute test accuracy for the baseline system and the baseline system combined with the specialist models. With 61 specialist models, there is a 4.4% relative improvement in test accuracy overall. We also report conditional test accuracy, which is the accuracy by only considering examples belonging to the specialist classes, and restricting our predictions to that subset of classes.

从已训练好的基线完整网络出发，专家模型的训练速度极快（在 JFT 数据集上仅需数日而非数周）。所有专家模型均完全独立进行训练。表 3 展示了基线系统及结合专家模型的基线系统的绝对测试准确率。使用 61 个专家模型后，整体测试准确率相对提升了 4.4%。我们还报告了条件测试准确率，即仅考虑属于专家类别样本、并将预测范围限定在该类别子集时的准确率。

# of specialists covering 涵盖的专家人数	# of test examples 测试样本数量	delta in top1 correct top1 正确率差值	relative accuracy change 相对准确率变化
0	**350037**	0	0.0%
1	**141993**	+1421	+3.4%
2	**67161**	+1572	+7.4%
3	**38801**	+1124	+8.8%
4	**26298**	+835	+10.5%
5	**16474**	+561	+11.1%
6	**10682**	+362	+11.3%
7	**7376**	+232	+12.8%
8	**4703**	+182	+13.6%
9	**4706**	+208	+16.6%
10 or more 十个或更多	**9082**	+324	+14.1%

Table 4: Top 1 accuracy improvement by # of specialist models covering correct class on the JFT test set.

表 4：在 JFT 测试集上，覆盖正确类别的专家模型数量对 Top 1 准确率的提升情况。

For our JFT specialist experiments, we trained 61 specialist models, each with 300 classes (plus the dustbin class). Because the sets of classes for the specialists are not disjoint, we often had multiple specialists covering a particular image class. Table 4 shows the number of test set examples, the change in the number of examples correct at position 1 when using the specialist(s), and the relative percentage improvement in top1 accuracy for the JFT dataset broken down by the number of specialists covering the class. We are encouraged by the general trend that accuracy improvements are larger when we have more specialists covering a particular class, since training independent specialist models is very easy to parallelize.

在我们针对 JFT 专业模型的实验中，我们训练了 61 个专业模型，每个模型包含 300 个类别（外加一个"垃圾桶"类别）。由于各专业模型的类别集合存在重叠，我们经常会有多个专业模型覆盖同一个图像类别。表 4 展示了测试集样本数量、使用专业模型后首位正确预测数量的变化，以及按类别覆盖专业模型数量细分的 JFT 数据集 top1 准确率相对提升百分比。我们欣喜地发现一个总体趋势：当有更多专业模型覆盖特定类别时，准确率提升更为显著，这得益于独立专业模型的训练能够轻松实现并行化。

6 Soft Targets as Regularizers

6 软目标作为正则化器

One of our main claims about using soft targets instead of hard targets is that a lot of helpful information can be carried in soft targets that could not possibly be encoded with a single hard target. In this section we demonstrate that this is a very large effect by using far less data to fit the 85M parameters of the baseline speech model described earlier. Table 5 shows that with only 3% of the data (about 20M examples), training the baseline model with hard targets leads to severe overfitting (we did early stopping, as the accuracy drops sharply after reaching 44.5%), whereas the same model trained with soft targets is able to recover almost all the information in the full training set (about 2% shy). It is even more remarkable to note that we did not have to do early stopping: the system with soft targets simply “converged” to 57%. This shows that soft targets are a very effective way of communicating the regularities discovered by a model trained on all of the data to another model.

我们关于使用软目标而非硬目标的主要观点之一是：软目标能够承载大量有用信息，这些信息无法通过单一硬目标进行编码。在本节中，我们通过使用极少量的数据来拟合先前描述的包含 8500 万参数的基线语音模型，证明了这种影响的显著性。表 5 显示，仅使用 3%的数据（约 2000 万个样本）时，采用硬目标训练的基线模型会出现严重过拟合（我们实施了早停策略，因为模型在达到 44.5%准确率后急剧下降），而使用软目标训练的相同模型几乎能完全恢复完整训练集中的信息（仅相差约 2%）。更值得注意的是，我们无需实施早停策略：采用软目标的系统直接"收敛"至 57%的准确率。这表明软目标是将全量数据训练模型所发现的规律有效传递给另一个模型的极佳方式。

System & training set **系统与训练集**	**Train Frame Accuracy 训练帧准确率**	**Test Frame Accuracy 测试框架准确率**
Baseline (100% of training set) **基线（100%训练集）**	63.4%	58.9%
Baseline (3% of training set) **基线（3%训练集）**	67.3%	44.5%
Soft Targets (3% of training set) **软目标（3%训练集）**	65.4%	57.0%

Table 5: Soft targets allow a new model to generalize well from only 3% of the training set. The soft targets are obtained by training on the full training set.

表 5：软目标使得新模型仅需训练集的 3%即可实现良好泛化。这些软目标是通过在完整训练集上训练获得的。

6.1 Using soft targets to prevent specialists from overfitting

6.1 使用软目标防止专家模型过拟合

The specialists that we used in our experiments on the JFT dataset collapsed all of their non-specialist classes into a single dustbin class. If we allow specialists to have a full softmax over all classes, there may be a much better way to prevent them overfitting than using early stopping. A specialist is trained on data that is highly enriched in its special classes. This means that the effective size of its training set is much smaller and it has a strong tendency to overfit on its special classes. This problem cannot be solved by making the specialist a lot smaller because then we lose the very helpful transfer effects we get from modeling all of the non-specialist classes.

在我们针对 JFT 数据集开展的实验中，专家模型将所有非专业类别合并为一个综合杂项类。若允许专家模型对所有类别进行完整的 softmax 运算，或许存在比早停法更有效的方法来防止其过拟合。专家模型是在其专业类别高度集中的数据上进行训练的，这意味着其有效训练集规模大幅缩减，极易对专业类别产生过拟合。单纯缩小专家模型规模无法解决此问题，因为这样做会使我们丧失通过建模所有非专业类别获得的宝贵迁移效应。

Our experiment using 3% of the speech data strongly suggests that 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. The soft targets can be provided by the generalist. We are currently exploring this approach.

我们使用 3%语音数据进行的实验充分表明：若采用通用模型的权重初始化专家模型，并通过同时使用硬目标与软目标进行训练（软目标可由通用模型提供），就能使其几乎完全保留对非专业类别的认知。我们目前正在深入探索这一方法。

7 Relationship to Mixtures of Experts

7 与专家混合模型的关系

The use of specialists that are trained on subsets of the data has some resemblance to mixtures of experts [6] which use a gating network to compute the probability of assigning each example to each expert. At the same time as the experts are learning to deal with the examples assigned to them, the gating network is learning to choose which experts to assign each example to based on the relative discriminative performance of the experts for that example. Using the discriminative performance of the experts to determine the learned assignments is much better than simply clustering the input vectors and assigning an expert to each cluster, but it makes the training hard to parallelize: First, the weighted training set for each expert keeps changing in a way that depends on all the other experts and second, the gating network needs to compare the performance of different experts on the same example to know how to revise its assignment probabilities. These difficulties have meant that mixtures of experts are rarely used in the regime where they might be most beneficial: tasks with huge datasets that contain distinctly different subsets.

使用在数据子集上训练的专家模型，与专家混合方法[6]存在相似之处，后者通过门控网络计算每个样本分配给各专家的概率。在专家学习处理所分配样本的同时，门控网络也根据各专家对特定样本的相对判别性能，学习如何选择专家进行分配。利用专家的判别性能来确定学习分配方案，远优于简单地对输入向量进行聚类并为每个聚类分配专家的做法，但这使得训练难以并行化：首先，每个专家的加权训练集会随着所有其他专家的状态持续变化；其次，门控网络需要比较不同专家对同一样本的表现，以调整其分配概率。这些困难导致专家混合方法在可能最受益的场景——包含明显不同子集的海量数据集任务中——鲜少被采用。

It is much easier to parallelize the training of multiple specialists. We first train a generalist model and then use the confusion matrix to define the subsets that the specialists are trained on. Once these subsets have been defined the specialists can be trained entirely independently. At test time we can use the predictions from the generalist model to decide which specialists are relevant and only these specialists need to be run.

并行训练多个专家模型要容易得多。我们首先训练一个通用模型，然后利用混淆矩阵确定各专家模型的训练子集。一旦这些子集确定后，专家模型便可完全独立进行训练。在测试阶段，我们可以根据通用模型的预测结果判断哪些专家模型与之相关，仅需运行这些相关的专家模型即可。

8 Discussion 8 讨论

We have shown that distilling works very well for transferring knowledge from an ensemble or from a large highly regularized model into a smaller, distilled model. On MNIST distillation works remarkably well even when the transfer set that is used to train the distilled model lacks any examples of one or more of the classes. For a deep acoustic model that is version of the one used by Android voice search, we have shown that nearly all of the improvement that is achieved by training an ensemble of deep neural nets can be distilled into a single neural net of the same size which is far easier to deploy.

我们已经证明，蒸馏法在将知识从集成模型或大型高度正则化模型迁移到更小的蒸馏模型方面效果显著。在 MNIST 数据集上，即使用于训练蒸馏模型的迁移集缺少一个或多个类别的任何样本，蒸馏依然表现出色。对于 Android 语音搜索所使用的深度声学模型的某个版本，我们证明了通过训练深度神经网络集成所获得的几乎所有性能提升，都可以被蒸馏到同等规模的单一神经网络中，这大大简化了部署过程。

For really big neural networks, it can be infeasible even to train a full ensemble, but we have shown that the performance of a single really big net that has been trained for a very long time can be significantly improved by learning a large number of specialist nets, each of which learns to discriminate between the classes in a highly confusable cluster. We have not yet shown that we can distill the knowledge in the specialists back into the single large net.

对于真正庞大的神经网络，训练完整集成模型甚至可能不可行，但我们已证明：通过训练大量专用网络——每个网络专门学习区分高度易混淆类别簇中的类别——可以显著提升经过长期训练的单一巨型网络的性能。目前我们尚未实现将专用网络中的知识蒸馏回单一大型网络的技术。

Acknowledgments 致谢

We thank Yangqing Jia for assistance with training models on ImageNet and Ilya Sutskever and Yoram Singer for helpful discussions.

我们感谢贾扬清在 ImageNet 上训练模型时提供的帮助，以及 Ilya Sutskever 和 Yoram Singer 的有益讨论。

References

[1] C. Buciluǎ, R. Caruana, and A. Niculescu-Mizil.
Model compression.
In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD ’06, pages 535–541, New York, NY, USA, 2006. ACM.

[2] J. Dean, G. S. Corrado, R. Monga, K. Chen, M. Devin, Q. V. Le, M. Z. Mao, M. Ranzato, A. Senior, P. Tucker, K. Yang, and A. Y. Ng.
Large scale distributed deep networks.
In *NIPS*, 2012.

[3] T. G. Dietterich.
Ensemble methods in machine learning.
In *Multiple classifier systems*, pages 1–15. Springer, 2000.

[4] G. E. Hinton, L. Deng, D. Yu, G. E. Dahl, A. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N Sainath, and B. Kingsbury.
Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups.
Signal Processing Magazine, IEEE, 29(6):82–97, 2012.

[5] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov.
Improving neural networks by preventing co-adaptation of feature detectors.
arXiv preprint arXiv:1207.0580, 2012.

[6] R. A. Jacobs, M. I. Jordan, S. J. Nowlan, and G. E. Hinton.
Adaptive mixtures of local experts.
Neural computation, 3(1):79–87, 1991.

[7] A. Krizhevsky, I. Sutskever, and G. E. Hinton.
Imagenet classification with deep convolutional neural networks.
In *Advances in Neural Information Processing Systems*, pages 1097–1105, 2012.

[8] J. Li, R. Zhao, J. Huang, and Y. Gong.
Learning small-size dnn with output-distribution-based criteria.
In *Proceedings Interspeech 2014*, pages 1910–1914, 2014.

[9] N. Srivastava, G.E. Hinton, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov.
Dropout: A simple way to prevent neural networks from overfitting.
The Journal of Machine Learning Research, 15(1):1929–1958, 2014.

◄ ar5iv

Feeling ◄ ar5iv 感觉 lucky?

Conversion 幸运? 转换 report (OK)

Report 报告问题 an issue

在 arXiv 上 ►

View original 查看原始问题 on arXiv

►

Copyright Privacy Policy Generated on Tue Mar 19 19:16:04 2024 by LaTeXML