Distilling the Knowledge in a Neural Network

Abstract

train many different models on the same data and then to average their predictions
but
cumbersome
too computationally expensive

we provide
improve the acoustic model of a heavily used commercial system
ensemble composed of one or more full models and many specialist models

Introduction

models are usually trained to optimize performance on the training data when the real objective is to generalize well to new data.
하지만 distilling에서는
we can train the small model to generalize in the same way as the large model

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
using a much higher learning rate.

원래 큰 모델 MNIST에서 very high confidence -> 학습에 좋지 않음
logits
“distillation” - temperature

using the original training set works well + a small term
-> to predict the true targets as well as matching the soft targets provided by the cumbersome mode;

Typically, the small model cannot exactly match the soft targets and erring in the direction of the correct answer turns out to be helpful.

Distillation

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

Using a higher value for T produces a softer probability distribution over classes.

simplest form of distillation
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

produce the correct labels
simply use a weighted average of two different objective functions
1 cross entropy with the soft targets from high t of the distilled model -> generating the soft targets from the cumbersome model
2 cross entropy with the correct labels -> exactly the same logits in softmax of the distilled model but at a temperature of 1
using a condiderably lower weight on the second objective function is good

it is important to multiply them by T^2 when using both hard and soft targets.
-> relative contributions of the hard and soft targets remain roughly unchanged

Matching logits is a special case of distillation

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

in the high temperature limit, distillation is equivalent to minimizing logits are zero-meaned separately for each transfer case
lower temperatures, distillation pays much less attention to matching logits that are much more negative than the average
when the distilled model is much too small to capture all of the knowledege in the cumbersome model, intermediate temperatures work best

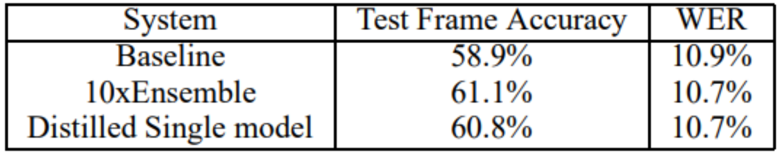
Preliminary experiments on MNIST

soft targets can transfer a great deal of knowledge to the distilled model
how to generalize that is learned from translated training data
with the right bias, training 때 한 번도 안 본 것도 잘 맞추더라

Experiments on speech recognition

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.

Results



System	Test Frame Accuracy	WER
Baseline	58.9%	10.9%
10xEnsemble	61.1%	10.7%
Distilled Single model	60.8%	10.7%

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.

Training ensembles of specialists on very big datasets

an ensemble of models is a very simple way to take advantage of parallel computation
but 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

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.

specialists that focus on making fine-grained distinctions is that they overfit very easily
-> prevented by using soft targets

The JFT Dataset

100 million labeled images with 15,000 labels.

Specialist Models

each of which is trained on data that is highly enriched in examples from a very confusable subset of the classes

single dustbin class.

half its examples coming from its special subset

half sampled at random from the remainder of the training set

correct for the biased training set by incrementing the logit of the dustbin class by the log of the proportion

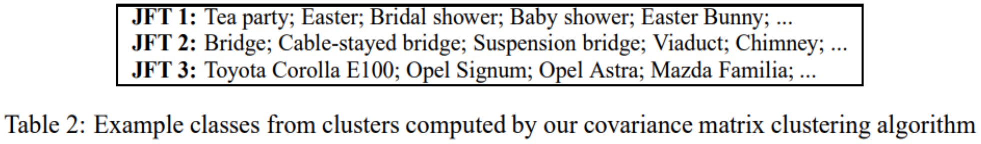
Assigning classes to specialists

simpler approach that does not require the true labels to construct the clusters

a clustering algorithm to the covariance matrix

a set of classes will be used as targets for one of our specialist models

on-line version of the K-means algorithm to the columns of the covariance matrix, and obtained reasonable clusters



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

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

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

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

Performing inference with ensembles of specialists

generalist model so that we can deal with classes for which we have no specialists

Step 1

n most probable classes according to the generalist model

n = 1

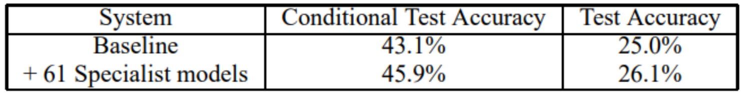
Step 2

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

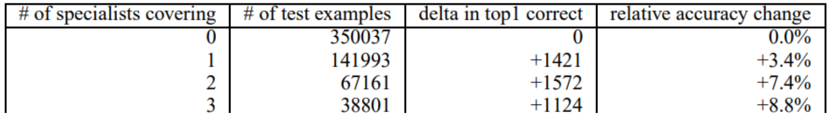
Results

the specialists train extremely fast

all the specialists are trained completely independently.



System	Conditional Test Accuracy	Test Accuracy
Baseline	43.1%	25.0%
+ 61 Specialist models	45.9%	26.1%



# of specialists covering	# of test examples	delta in top1 correct	relative accuracy change
0	350037	0	0.0%
1	141993	+1421	+3.4%
2	67161	+1572	+7.4%
3	38801	+1124	+8.8%

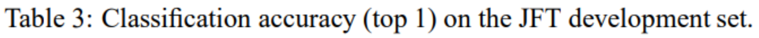
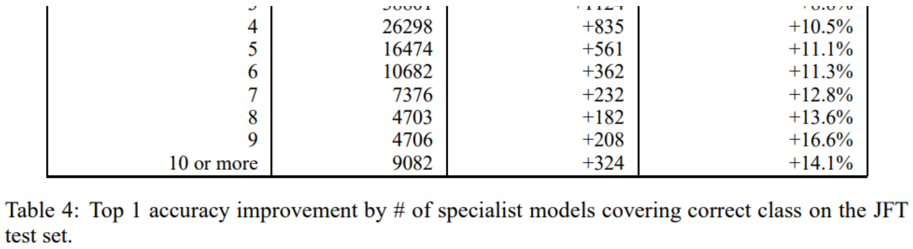


Table 3: Classification accuracy (top 1) on the JFT development set.



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	9082	+324	+14.1%

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

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

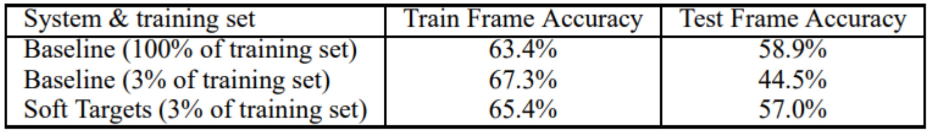
We are encouraged by the general trend that accuracy improvements are larger when we have more specialists covering a particular class,

Soft Targets as Regularizers

a lot of helpful information can be carried in soft targets
this is a very large effect by using far less data to fit the 85M parameters of the baseline speech model described earlier

training the baseline model with hard targets leads to severe overfitting
whereas the same model trained with soft targets is able to recover almost all the information in the full training set
we did not have to do early stopping: the system with soft targets simply “converged” to 57%

soft targets are a very effective way of communicating the regularities discovered by a model trained on all of the data to another model



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%

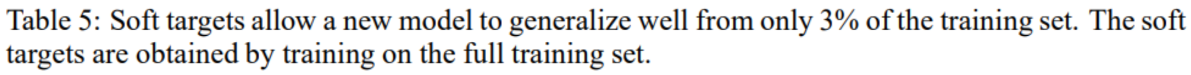


Table 5: Soft targets allow a new model to generalize well from only 3% of the training set. The soft targets are obtained by training on the full training set.

Using soft targets to prevent specialists from overfitting

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
effective size of its training set is much smaller
it has a strong tendency to overfit
-> cannot be solved by making the specialist a lot smaller because then we lose the very helpful transfer effects

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

Relationship to Mixtures of Experts

gating network to compute the probability of assigning each example to each expert

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

but it makes the training hard to parallelize

1 the weighted training set for each expert keeps changing in a way that depends on all the other experts

2 the gating network needs to compare the performance of different experts on the same example to know how to revise its assignment probabilities

따라서

We first train a generalist model and then use the confusion matrix to define the subsets that the specialists are trained on

specialists can be trained entirely independently

predictions from the generalist model to decide which specialists are relevant

only these specialists need to be run

Discussion

distilling works very well for transferring knowledge from an ensemble or from a large highly regularized model into a smaller, distilled model

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

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

We have not yet shown that we can distill the knowledge in the specialists back into the single large net