Distilling the Knowledge in a Neural Network

G. Hinton, O. Vinyals, J. Dean

Google Inc.

arXiv: 1807.10875

Reviewed by : Bill Zhang
University of Virginia
https://qdata.github.io/deep2Read/

Outline

Introduction

Basic Premise and Motivation

- Deployment of ML models to large number of users has restrictions on latency and computational resources
- Can transfer knowledge from more cumbersome model to smaller model
- Look at ML models as mappings from input to output vectors
- Transfer both accuracy and generalization by looking also at the relative distribution of wrong classes; use class probabilities of large model as soft target for small models
- Raise temperature of softmax until targets soft enough
- Cannot exactly match soft targets, but erring in direction of correct answer produces good results

Distillation

- Neural networks typically produce a softmax layer which converts logits to probability with some temperature T usually set to 1
$$q_i = \frac{\exp(z_i/T)}{\sum_j \exp(z_j/T)}$$
- In simplest form, distillation is performed by training distilled model on transfer set using a soft target distribution produced by large model with high temperature in its softmax
- Best results when training on two weighted objective functions: cross entropy with soft targets and cross entropy with correct labels, with small weight on the second

Distillation

Matching Logits

- Through derivations, arrive at expression

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

  where z_i are logits of distilled model and v_i are logits of cumbersome model
- In high temperature limit, distillation equivalent to minimizing $1/2(z_i - v_i)^2$
- At lower temperatures, distillation pays less attention to matching logits much more negative than average; could be advantageous (remove noise) or disadvantageous (remove important generalization information)
- Empirically determine temperature to be some intermediate value

Preliminary Experiments on MNIST

- Trained single large neural net with 2 hidden layers of 1200 ReLUs each on all 60,000 training cases
- Regularized using dropout and weight-constraints, images jittered in 2 pixels in any direction
- Large model produced 67 errors
- Small model with 2 hidden layers of 800 ReLUs each achieved 146 errors, but lowered number to 74 when training while trying to match large model soft targets (with $T = 20$)
- Temperature can be empirically altered
- Omitting all examples of 3 on the transfer set only increased error count to 206, 133 of which were the 3s

Experiments on Speech Recognition

- Investigate effects of ensembling DNN acoustic models using in Automatic Speech Recognition (ASR); show that we can distill ensemble into 1 model of same size as other individual models
- State-of-the-art ASR systems map a short temporal context of features from the waveform to a probability distribution over all states of a Hidden Markov Model (HMM)
- Use architecture with 8 hidden layers each containing 2,560 ReLUs and final softmax with 14,000 labels; input is 26 frames of 40 Mel-scaled filter-bank coefficients with 10ms advance per frame
- Predict HMM state of 21st frame

Experiments on Speech Recognition

Results

- Train 10 separate models with exact same architecture and training procedure as baseline; random initialization
- Varying training data did not significantly change results
- For distillation, tried temperatures of 1, 2, 5, 10 and used 0.5 for relative weight of hard target cross entropy

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

Training Ensembles of Specialists

- Training ensembles is effective because we can take advantage of parallelization
- However, in some cases, even parallelization is not enough if the dataset is large enough
- Thus, explore how specialist models can be used to cut computational costs for these cases

Training Ensembles of Specialists

JFT Dataset

- The JFT dataset is an internal Google dataset with 100 million labeled images with 15,000 labels
- Google baseline model is deep CNN trained for around 6 months with multiple cores
- Two types of parallelism: (1) multiple replicas trained on multiple cores and processing mini-batches from training set communicating with shared parameter server and (2) Each replica spread over multiple cores
- Ensembling can be wrapped around these methods provided there are more cores
- Needed a faster way to improve baseline

Training Ensembles of Specialists

Specialist Models

- Makes sense to have one generalist model and many "specialist" models which are trained on data highly enriched in examples from a very confusable subset of classes: examples include different types of mushrooms
- For specialist models, every class that does not matter can be combined into a dustbin class
- To reduce overfitting, each specialist initialized with parameters of generalist model
- Then, take half of examples from special subset and other half from remainder of training set; in the end, to account for bias, increment dustbin logit by log of proportion by which specialist class oversampled

Training Ensembles of Specialists

Assigning Classes

- Focus on categories which full network often confuses
- Instead of using confusion matrices to cluster, apply clustering algorithm to covariance matrix of the predictions of our generalist model
- This is done such that a set of classes S^m that are often predicted together will be used as targets for specialist model m

Training Ensembles of Specialists

Inferences with Ensemble of Specialists

- Given input image X, do top-one classification in two steps
- (1) For each test case, find n = 1 most probable classes according to generalist model
- (2) Take all specialist models with non-empty intersection with the n selected classes as set A_k
- Find full probability distribution q which minimizes

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

  where KL is the KL divergence
- If a single probability is produced for each class, can just take arithmetic or geometric mean
- Parameterize $q = softmax(z)$ with $T = 1$ and use gradient descent to optimize z

Training Ensemble of Specialists

Results

- Initialization allows specialist models to train in a few days instead of months
- Specialist models, overall, do seem to improve model accuracy

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

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

# 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%
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.

Soft Targets as Regularizers

- With limited data (3% of data), training on hard targets resulted in severe overfitting; had to stop early
- Training on soft targets allowed model to reach within 2% of baseline accuracy; did not require early stopping
- Might have been better to have specialists train with full number of classes instead of using a dustbin class to prevent overfitting

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.

Relationship to Mixtures of Experts

- Training using specialists is similar to training using experts which use a gating network to compute probability of assigning each example to an expert
- Gating network learns to choose which experts to assign examples to using the relative discriminative performance of the experts for that example
- Main problem is difficulties with parallelizing this process since assignment probabilities depend on all the experts
- Much easier to parallelize training of specialist models

Discussion

- Showed that distillation is very effective for transferring knowledge from large, highly regularized model to smaller, distilled model
- On MNIST, distillation works well even when transfer set lacks some classes
- On deep acoustic models, distillation of entire ensemble into single model works well
- For large enough models, an ensemble maybe infeasible, but the performance can be improved by training specialist nets; as of now, have not yet shown that these can be distilled into one final net

References

- https://arxiv.org/pdf/1503.02531.pdf