Knowledge Distillation

Distilling the Knowledge in a Neural Network (2015) [1]
G. Hinton, O. Vinyals, J. Dean

UBC CPSC 532S   Mar 28, 2019

Farnoosh Javadi
Jiefei Li
Si Yi (Cathy) Meng
Muhammad Shayan

Motivation

Motivation

- Improving the performance of machine learning algorithms
  - Ensemble method
    - Cumbersome
    - Computationally expensive

Motivation

- Improving the performance of machine learning algorithms
  - Ensemble method
    - Cumbersome
    - Computationally expensive
  - Distillation
    - Compressing the knowledge of a cumbersome model into a single small model

Motivation

- Improving the performance of machine learning algorithms
    - Ensemble method
        - Cumbersome
        - Computationally expensive
    - Distillation
        - Compressing the knowledge of a cumbersome model into a single small model
    - Cumbersome model
        - Ensemble of many models
        - Single complex huge model

Intuition

- Model's knowledge
    - Learned parameters of the model
        - Hard to transfer when we change the form of model

Intuition

- Model's knowledge
    - Learned parameters of the model
        - Hard to transfer when we change the form of model
    - Learned mapping from input vectors to output vectors
        - Frees it from dependency to model structure

Intuition

- Model’s knowledge
    - Learned parameters of the model
        - Hard to transfer when we change the form of model
    - Learned mapping from input vectors to output vectors
        - Frees it from dependency to model structure
            - Cumbersome classifier
                - Maximizes the probability of correct class
                - Assigns small weights to incorrect classes which in spite of being small are informative
                    - Mistaking BMW as a truck is much more that a carrot

Method Overview

Method overview

- Transfer cumbersome model’s knowledge
  - Use predictions of the cumbersome model as soft targets of simple model

Method overview

- Transfer cumbersome model’s knowledge
  - Use predictions of the cumbersome model as soft targets of simple model
- Advantages
  - Small model is more general
  - Training needs less data
  - Training is faster

Method overview

- Transfer cumbersome model's knowledge
  - Use predictions of the cumbersome model as soft targets of simple model
- Advantages
  - Small model is more general
  - Training needs less data
  - Training is faster
- Disadvantage
  - Weights for incorrect classes are too small to influence in the gradient

Method overview

- Transfer cumbersome model’s knowledge
    - Use predictions of the cumbersome model as soft targets of simple model
- Advantages
    - Small model is more general
    - Training needs less data
    - Training is faster
- Disadvantage
    - Weights for incorrect classes are too small to influence in the gradient
        - Solution
            - Previous work of Caruana [5]: Use logits as soft targets

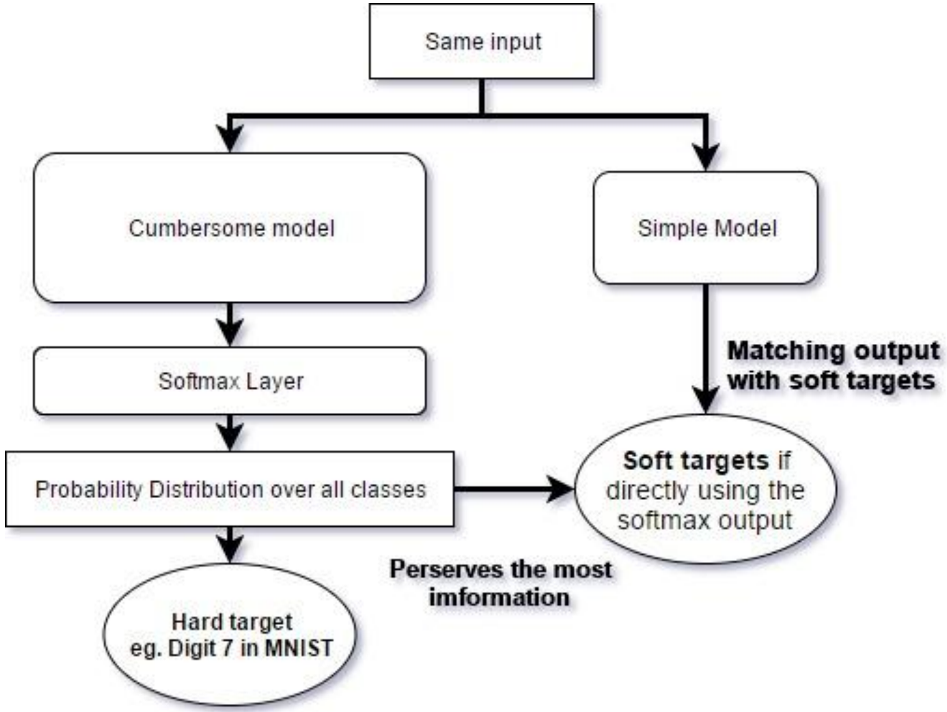
Method overview

- Transfer cumbersome model's knowledge
  - Use predictions of the cumbersome model as soft targets of simple model
- Advantages
  - Small model is more general
  - Training needs less data
  - Training is faster
- Disadvantage
  - Weights for incorrect classes are too small to influence in the gradient
    - Solution
      - Previous work of Caruana [5]: Use logits as soft targets
      - This paper: Raise temperature of softmax to produce suitable soft targets

Methods & Experiments

How does distillation work?

- Train a simple model on a transfer set.
- In the transfer set, the data labels are the soft target distribution produced by the cumbersome model trained with a high temperature value.



Temperature

A hyperparameter that controls amount of scaling the logits.

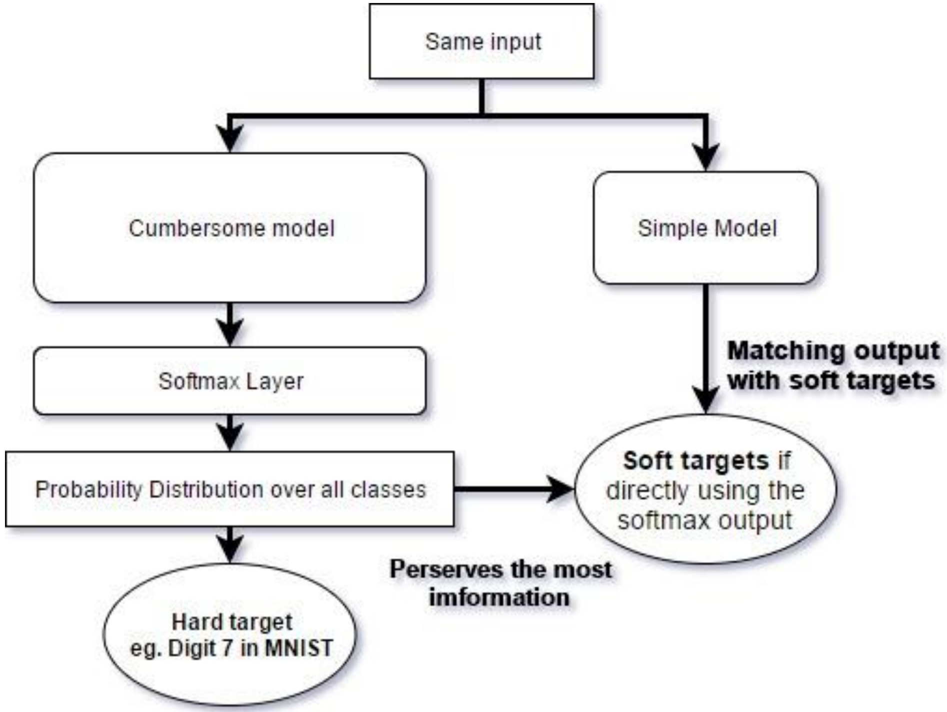
$$q_i = \frac{exp(z_i/T)}{\sum_j exp(z_j/T)}$$

[1]

- Normal softmax: T=1 (Compute the softmax directly on the logits)
- A higher value for T produces softer probability distribution over classes.
- (Softer probability distribution refers to more uniform-distributed probability)

How does distillation work?

- Train a simple model on a transfer set.
- In the transfer set, the data labels are the soft target distribution produced by the cumbersome model trained with a high temperature value.
    - The same high T value will also be used in the simple model's softmax during its training.
    - After training, set T to 1.



Use the distilled model to predict the correct labels

Predict the correct labels using the weighted average of two different objective functions

- ➔ The cross entropy with the soft targets, computed using the same high temperature value.
- ➔ The cross entropy with the hard targets, computed using a temperature value of 1.
- ➔ Higher weight for the first objective function and considerably low weight for the second objective function.
- ➔ Important to multiply the magnitude of gradient produced by the soft target by T^2

Matching Logits in distillation

Cross-Entropy gradient of each case in the transfer set is given by

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

If temperature is high compared with the magnitude of the logits

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

The logits have been zero-meaned for each transfer case

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

[1]

Where z_i is a logit in the distilled model, v_i is a logit in the cumbersome model, p_i is the soft target generated by the cumbersome model.

Matching Logits in distillation

With the high temperature limit, through approximation and simplifying, we found:

Minimizing the cross-entropy of the distilled model is equivalent to minimizing

$$\frac{1}{2}(z_i - v_i)^2$$

At lower temperature, logits much more negative than the average will be ignored when matching logits.

Why it is potentially advantageous?

For small-size distilled model, intermediate temperature works the best.

Preliminary MNIST experiments

Cumbersome Model

- 2 hidden layers
- 1200 rectified linear units
- Trained with dropout and weight constraints

Distilled Model

- 2 hidden layers
- 800 rectified linear units
- No regularization

Cumbersome model trained on original dataset => 67 test errors

Distilled model trained on original dataset => 146 errors

Distilled model trained by matching soft targets of cumbersome model =>74 test errors

Preliminary MNIST experiments

Cumbersome Model

- 2 hidden layers
- 1200 rectified linear units
- Trained with dropout and weight constraints

Distilled Model

- 2 hidden layers
- 800 rectified linear units
- No regularization

All temperatures above 8 gave a fairly similar result with 300 units in the distilled model

Even if a digit is omitted in the transfer set, the distilled model still learns to recognize it even if the learned bias on the cumbersome model is high

Distilling an ensemble of models into a single model

- Train multiple separate models to predict the same probability distributions.
- Ensemble the prediction from all models to create the soft target for training the simple model.
- Train the simple model with the average predictions from the ensemble.
- The temperature value used both in the ensemble and the single model needs being tuned to find the best value.

Speech Recognition Experiments

Speech Recognition DNN trained on 2000 hours of spoken English data

Baseline DNN => 58.9 % accuracy

10x Ensemble => 61.1% accuracy

Distilled Single Model => 60.8% accuracy

More than 80% of improvement in frame accuracy achieved by ensemble transferred to distilled model

Ensembles of specialist models

Ensemble is great, but training an army of DNNs can be intractable.

Ensembles of specialist models

Ensemble is great, but training an army of DNNs can be intractable.

Solution: Divide a large number of classes into multiple confusable subsets of the classes, and train specialist models on small portion of data that belong to each class subset.

Ensembles of specialist models

Ensemble is great, but training an army of DNNs can be intractable.

Solution: Divide a large number of classes into multiple confusable subsets of the classes, and train specialist models on small portion of data that belong to each class subset.

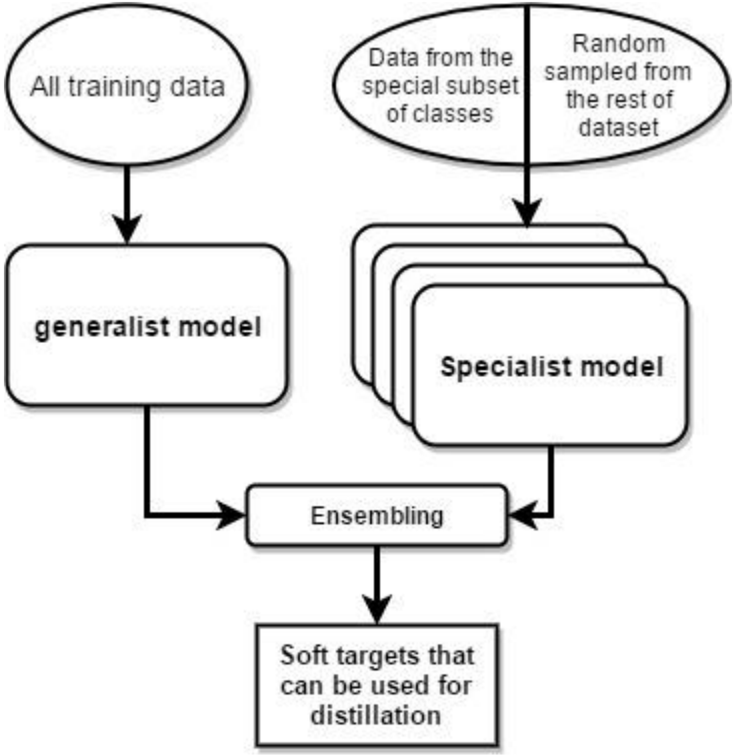
Pro:

- Requires less training time since dataset is smaller for each specialist model.
- The softmax of this type of specialist model is smaller, since it combines all of the classes don’t belong to its class subset into a dustbin class.

Con:

- Each specialist model may get overfitted easily.

The structure of ensembles of specialists



Generalist model handles all classes for which don’t have specialists.

Each Specialist model handles a subset of classes for which are usually predicted together.

Specialist modes are initiated with the weights of the generalist model.

How to assign classes to specialists

- Apply a clustering method (online version of K-means) to the covariance matrix of the predictions of the generalist model.
- A set of classes which are often predicted together are assigned to one of the specialists.

Inference with ensembles of specialists

1. Use the generalist model to pick the set of most probable classes to be the class set K.
2. Pick all the specialist models whose special class subset has a non-empty intersection with K.
3. Then find a full probability distribution q that can minimize the target function

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

[1]

4. The solution to the above equation is either the arithmetic or geometric mean of predictions from specialist models.

The full distribution q is considered as the result of softmax of logit Zs.

Training specialist ensembles on big datasets

Training an ensemble can potentially lead to a accuracy but requires a lot more compute resources to train in parallel.

Specialist Ensembles can be trained quickly by distilling the cumbersome model.

61 distilled specialist ensembles lead to a 1.1% accuracy

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

[1]

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

Soft targets as regularizers

Training on soft targets instead of hard targets leads to the model generalizing well.

Soft targets didn't even need early stopping while the baseline did.

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%

[1]

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.

Conclusion & Remarks

Main contributions / Strengths

- Distillation
- Transfer set can be any compatible dataset
    - Labels not necessary
- Equivalence with matching logits at high temperature
- Training specialists as ensemble

Weaknesses / Unresolved questions

- Distilling into a smaller network with a different architecture?

Weaknesses / Unresolved questions

- Distilling into a smaller network with a different architecture?
- What if the final layer isn't softmax?
  - Does heating it up still make sense?

Weaknesses / Unresolved questions

- Distilling into a smaller network with a different architecture?
- What if the final layer isn't softmax?
    - Does heating it up still make sense?
- Reverse distillation
    - Can we distill the knowledge in the specialists back into the single large model?

Weaknesses / Unresolved questions

- Distilling into a smaller network with a different architecture?
- What if the final layer isn't softmax?
  - Does heating it up still make sense?
- Reverse distillation
  - Can we distill the knowledge in the specialists back into the single large model?
- Distillation from specialists
  - The goal is to use ensemble for inference, using specialists still require having a generalist model

Weaknesses / Unresolved questions

- Distilling into a smaller network with a different architecture?
- What if the final layer isn't softmax?
  - Does heating it up still make sense?
- Reverse distillation
  - Can we distill the knowledge in the specialists back into the single large model?
- Distillation from specialists
  - The goal is to use ensemble for inference, using specialists still require having a generalist model
- Can we use the distilled model as a feature extractor?

Other approaches

- **MobileNets** [2, 3]
    - A family of mobile-first computer vision models
    - Maximize accuracy with low-latency, low-power/memory consumption
    - How? **Depth-wise separable filters**
        - Factorize a standard convolution into a depthwise convolution and a 1×1 pointwise convolution
        - Significantly reduces the number of parameters with minimal sacrifice in accuracy
    - **Distilling** a cumbersome model **into a MobileNet** architecture demonstrates enhanced performance

Other approaches

- **MobileNets** [2, 3]
  - A family of mobile-first computer vision models
  - Maximize accuracy with low-latency, low-power/memory consumption
  - How? **Depth-wise separable filters**
    - Factorize a standard convolution into a depthwise convolution and a 1×1 pointwise convolution
    - Significantly reduces the number of parameters with minimal sacrifice in accuracy
  - **Distilling** a cumbersome model **into a MobileNet** architecture demonstrates enhanced performance
- **Distillation and Quantization** [4]: two compression methods
  - Quantized distillation
  - Differentiable quantization

References

1. Hinton G, Vinyals O, Dean J. ***Distilling the knowledge in a neural network.*** arXiv preprint arXiv:1503.02531. 2015 Mar 9.
2. Howard AG, Zhu M, Chen B, Kalenichenko D, Wang W, Weyand T, Andreetto M, Adam H. ***Mobilenets: Efficient convolutional neural networks for mobile vision applications.*** arXiv preprint arXiv:1704.04861. 2017 Apr 17.
3. Howard AG and Zhu M. ***MobileNets***. 2017.
4. Polino A, Pascanu R, Alistarh D. ***Model compression via distillation and quantization.*** arXiv preprint arXiv:1802.05668. 2018 Feb 15.
5. Buciluǎ C, Caruana R, Niculescu-Mizil A. ***Model compression.*** InProceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining 2006 Aug 20 (pp. 535-541). ACM.

Thank you