Deep Model Compression

Xin Wang

Oct.31.2016

Some of the contents are borrowed from Hinton's and Song's slides.

Two papers

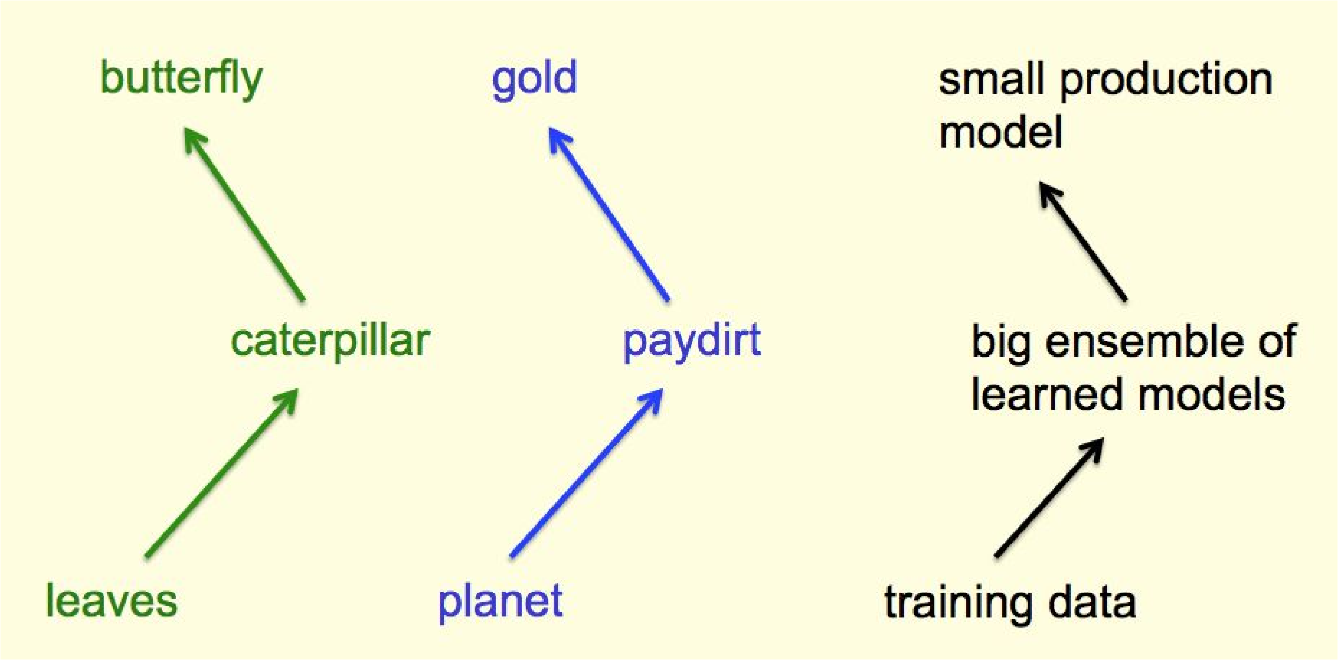
- Distilling the Knowledge in a Neural Network by Geoffrey Hinton et al
  - What’s the “dark” knowledge of the big neural networks?
  - How to transfer knowledge from big general model(teacher) to small specialist models(student)?

- Deep Compression: Compressing Deep Neural Networks with Pruning, Trained Quantization and Huffman Coding by Song Han et al
  - Provide a systematic way to compress big deep models.
  - The goal is to reduce the size of models without losing any accuracy.

The Conflicting Constraints of Training and Testing

- Training Phase:
  - The easiest way to extract a lot of knowledge from the training data is to learn many different models in parallel.
  - 3B: Big Data, Big Model, Big Ensemble
  - Imagenet: 1.2 million pictures in 1,000 categories.
  - AlexNet: ~ 240Mb, VGG16: ~550Mb
- Testing Phase:
  - Want small and specialist models.
  - Minimize the amount of computation and the memory footprint.
  - Real time prediction
  - Even able to run on mobile devices.

Knowledge Transfer: an Analogy



Knowledge Transfer: Main Idea

- Introduce “Soft targets” as a way to transfer the knowledge from big models.
    - classifiers built from a softmax function have a great deal more information contained in them than just a classifier;
    - the correlations in the softmax outputs are very informative.
- Caruana et. al. 2006 had the same idea but used a different way of transferring the knowledge
    - Direct match the logits (distribution over all the categories)
    - Hinton’s paper shows this is a special case of the “soft targets”

Hard Targets vs Soft Targets

- Hard Target: the ground truth label (one-hot vector)
- Soft Target: T is “temperature”, z is logit

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

- More information in soft targets

cow	dog	cat	car	
0	1	0	0	original hard targets

cow	dog	cat	car	
.05	.3	.2	.005	softened output of ensemble

Match both Soft Targets and Hard Targets

- Learn logits in the distilled model that minimize the sum of two different cross entropies
- Using a high temperature in the softmax, we minimize the cross entropy with the soft targets derived from the ensemble at high temperature
- Using the very same logits at a temperature of 1, we minimize the cross entropy with the hard targets.
- Relative weighting of the hard and soft cross entropies
  - The derivatives for the soft targets tend to be much smaller.
  - Down-weight the cross entropy with the hard targets.

Case 1: Train small model on the same dataset

- Experiment on MNIST
    - Vanilla backprop in a 784 -> 800 -> 800 -> 10 net with rectified linear hidden units (y=max(0,x)) gives 146 test errors. (10k test cases)
    - Train a 784 -> 1200 -> 1200 -> 10 net using dropout and weight constraints and jittering the input (add noise), get 67 errors.
    - Using both the soft targets obtained from the big net and the hard targets, we get 74 errors in the 784 -> 800 -> 800 -> 10 net.

Case 2: Train small model on small dataset

- Experiment on MNIST:
  - Train the 784 -> 800 -> 800 -> 10 net on a transfer set that does not contain any examples of a 3. After this training, raise the bias of the 3 by the right amount.
    - The distilled net then gets 98.6% of the test threes correct even though it never saw any threes during the transfer training.
  - Train the 784 -> 800 -> 800 -> 10 net on a transfer set that only contains images of 7 and 8. After training, lower the biases of 7 and 8 by the optimal amount.
    - The net then gets 87% correct over all classes.
- Similar results got on a Google's production speech model. (get 6/7 of the ensemble accuracy when training on only 3% of the dataset)

Ensemble Model & Specialist Nets

- To make an ensemble mine knowledge more efficiently, we encourage different members of the ensemble to focus on resolving different confusions.
    - In ImageNet, one “specialist” net could see examples that are enriched in mushrooms.
    - Another specialist net could see examples enriched in sports cars.
- K-means clustering on the soft target vectors produced by a generalist model works nicely to choose the confusable classes.
- Problems with Specialists
    - Specialists tend to over-fit.

One way to prevent specialists overfitting

- Each specialist uses a reduced softmax that has one dustbin class for all the classes it does not specialize in.
- The specialist estimates two things:
  - Is this image in my special subset?
  - What are the relative probabilities of the classes in my special subset?
- After training we can adjust the logit of the dustbin class to allow for the data enrichment.
- The specialist is initialized with the weights of a previously trained generalist model and uses early stopping to prevent over-fitting.

Experiment on JFT dataset

- This is a Google internal dataset with about 100 million images with 15,000 different class labels. (much larger than ImageNet)
- Takes 6 months to train one model with a lot of machines. (unrealistic to train dozens of models for ensembling)
- The baseline model has 25% test top-1 accuracy.
- Training an ensemble of 61 specialist, the model has 26.1% top-1 accuracy. (4.4% relative improvement)
- Also find accuracy improvements are larger when having more specialists covering a particular class.

Soft Targets as Regularizers

- Each specialist gets data that is very enriched in its particular subset of classes but its softmax covers all of the classes.
- On data from its special subset (50% of its training cases) it just tries to fit the hard targets with T=1.
- On the remaining data it just tries to match the soft targets produced by a previously trained generalist model at high temperature.
- Recall the speech model experiment with only 3% training data --- soft targets prevent overfitting.
- The authors didn’t provide any experimental results about this ensemble in the paper.

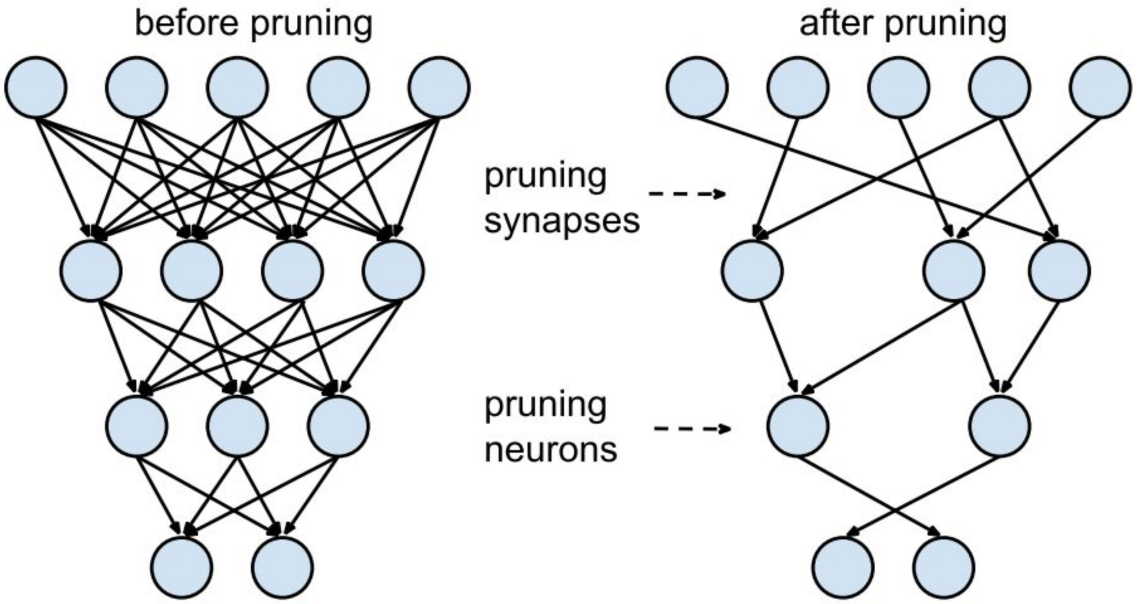
Conclusion of Distillation Paper

- A solution the previous mentioned *conflicting constraints of training and testing*
    - When extracting knowledge from data we do not need to worry about using very big models or very big ensembles of models that are much too cumbersome to deploy.
    - If we can extract the knowledge from the data it is quite easy to distill most of it into a much smaller model for deployment.
- Speculation:
    - On really big datasets, ensembles of specialists should be more efficient at extracting the knowledge.
    - Soft targets for their non-special classes can be used to prevent them from over-fitting.

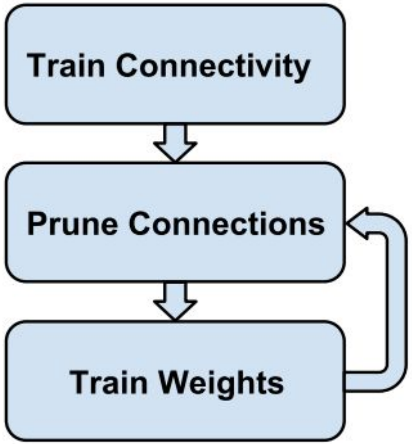
Deep Compression: A systematic way

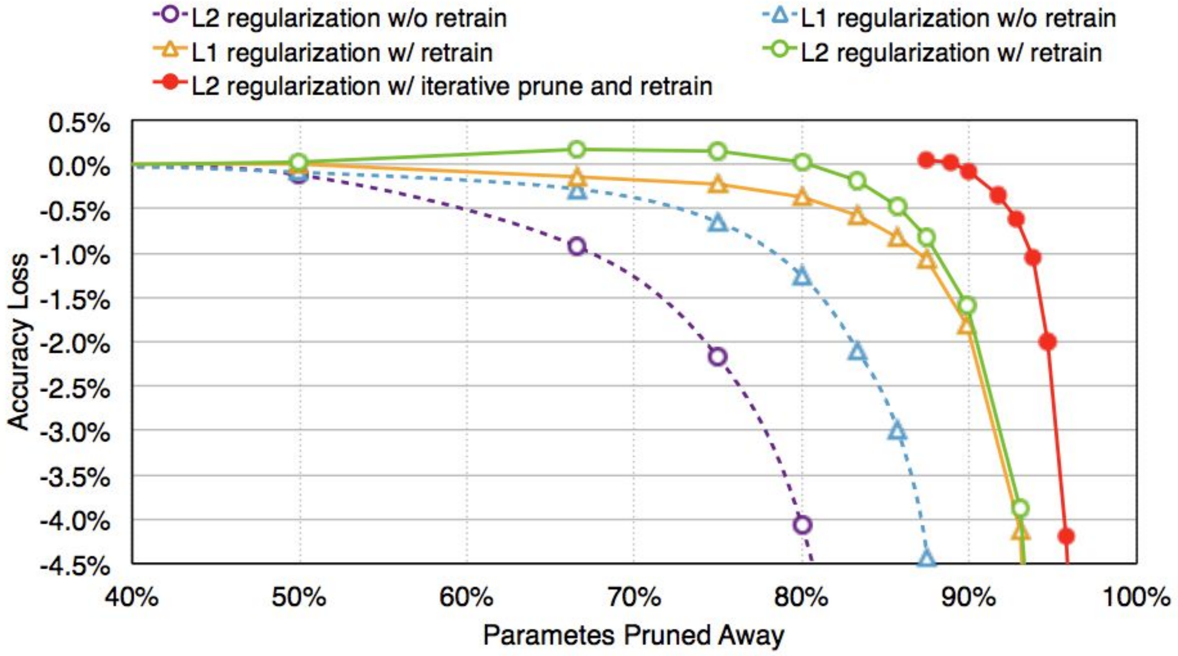
- The distillation paper provides a way to train small model inheriting from big general model.
  - Extract knowledge of the big models
- Deep Compression paper uses a pipeline: pruning, quantization and huffman coding to compress the models.
  - Directly do the surgery on the big models

Pruning


before pruning
after pruning
pruning
synapses
pruning
neurons

Retrain to Recover Accuracy





Network pruning can save 9x to 13x parameters without drop in accuracy

Weight Sharing (Trained Quantization)

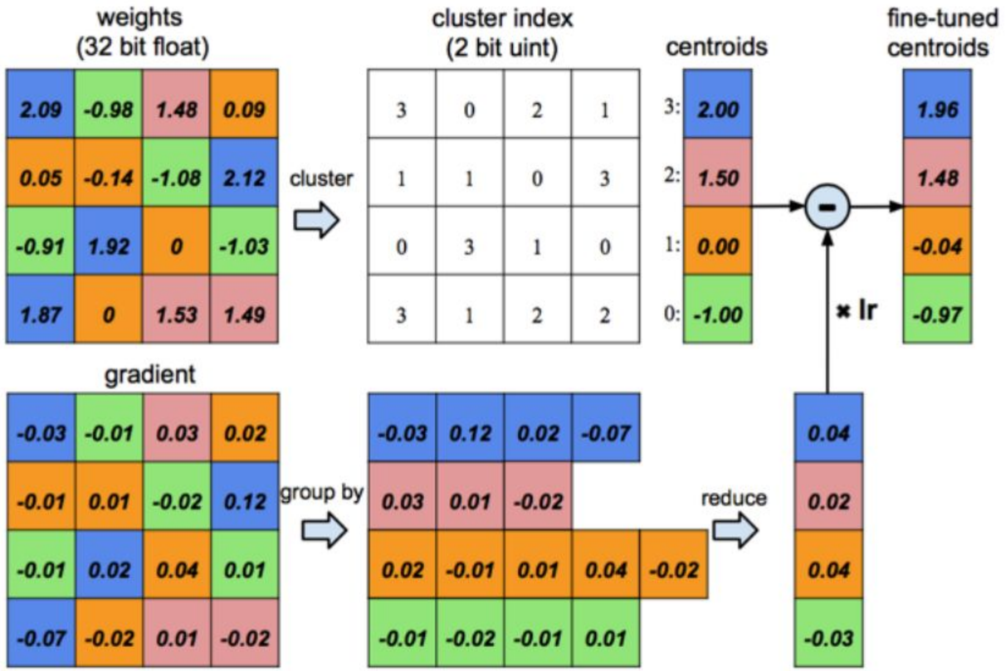
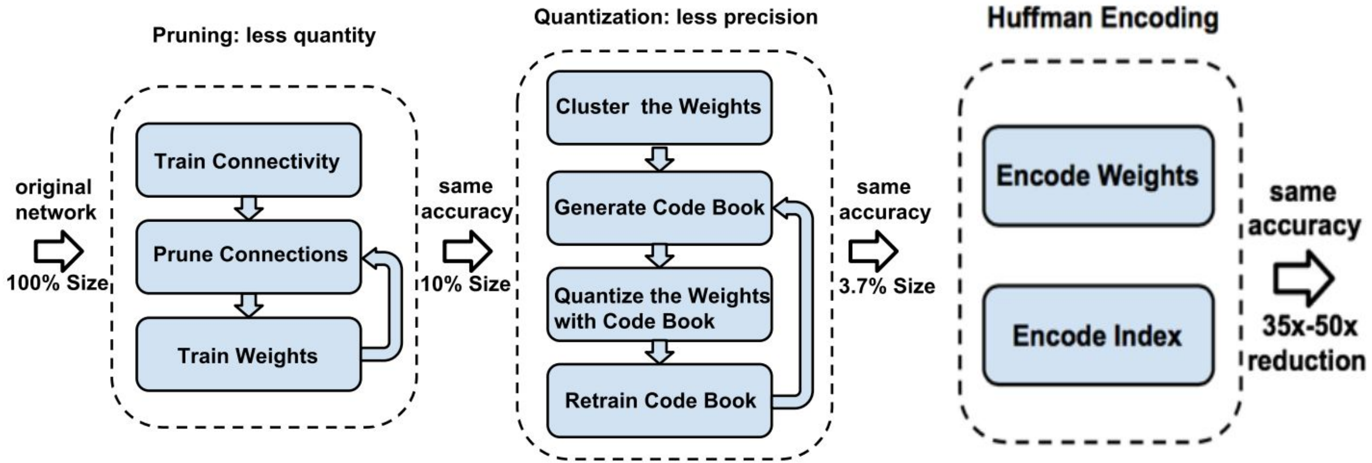


Figure 3: Weight sharing by scalar quantization (top) and centroids fine-tuning (bottom)

Huffman Coding



Results Highlight

- AlexNet: 35×, 240MB => 6.9MB => 0.52MB
- VGG16: 49× 552MB => 11.3MB
- Both with no loss of accuracy on ImageNet12

The End