Lecture 9: Compression Techniques and Distillation

AC295

Advanced Practical Data Science

Pavlos Protopapas

Outline

1: Communications

2: Motivations and what is Compression

3: Compression Techniques

4: Distillation

Communications

- Exercise 7 was due 10:15 AM.
- Exercise 8 will be released today - due (11/10 10:15 AM)
- Reading questions due tomorrow 11/04 noon on Ed.
- Last set of presentations this Thursday - 11/05
- Practicum will be released by Sunday - due 11/17
- Practicum week - No lectures on 11/10 Tue and 11/12 Thursday
- Three lectures remaining in the semester - 11/17, 11/19 and 11/24

Why do we need it?

We want to process data (ideally a lot) and we do not have enough computing resources. For example:

1. your phone can't run GoogleNet to assist you in some tasks
2. you can't compress ginormous images coming from the space (8Kx8K pixels from 3K satellites)

Using machine learning is resource intensive:

i. computing power to train M/B parameters

ii. limited bandwidth (you could use)

So what? Model compression techniques

Hannah Peterson and George Williams, *An Overview of Model Compression Techniques for Deep Learning in Space*, August 2020

What is Model Compression?

The main idea is to simplify the model without diminishing accuracy. A simplified model means reduced in size and/or latency from the original. Both types of reduction are desirable.

- Size reduction can be achieved by reducing the model parameters and thus using less RAM.
- Latency reduction can be achieved by decreasing the time it takes for the model to make a prediction, and thus lowering energy consumption at runtime (and carbon footprint).

Karen Hao, *Training a single AI model can emit as much carbon as five cars in their lifetimes*, June 2019

Compression Techniques (Algos)

1. Pruning
2. Quantization
3. Low-rank approximation and sparsity
4. Knowledge distillation
5. Low-rank approximation and sparsity
6. Neural Architecture Search (NAS) [another class]

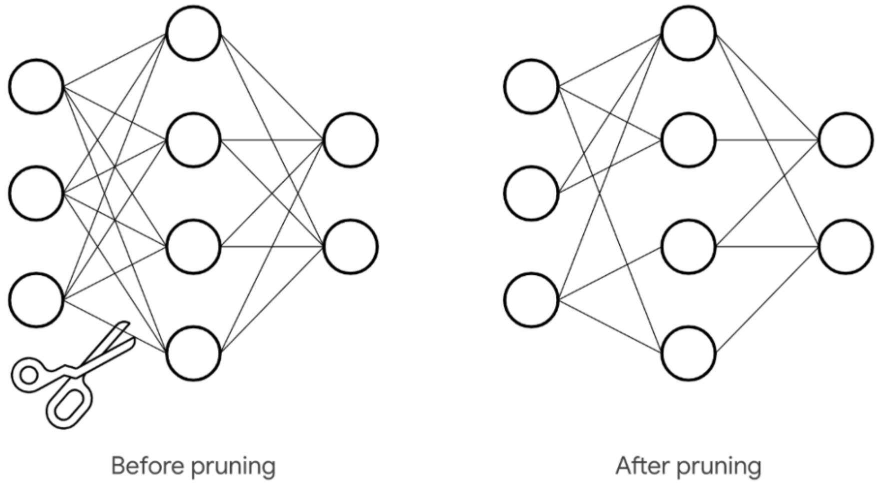
Compression Techniques: Pruning

The main idea is to remove features with nearly the same information.

Pruning involves removing connections between neurons, channels, or filters from a trained network. To prune a connection, we set a weight in the matrix to zero. To prune a neuron, we set all values of a column to zero.

2 types of pruning:

- Unstructured removes connections or neurons
- Structured removes filters or channels

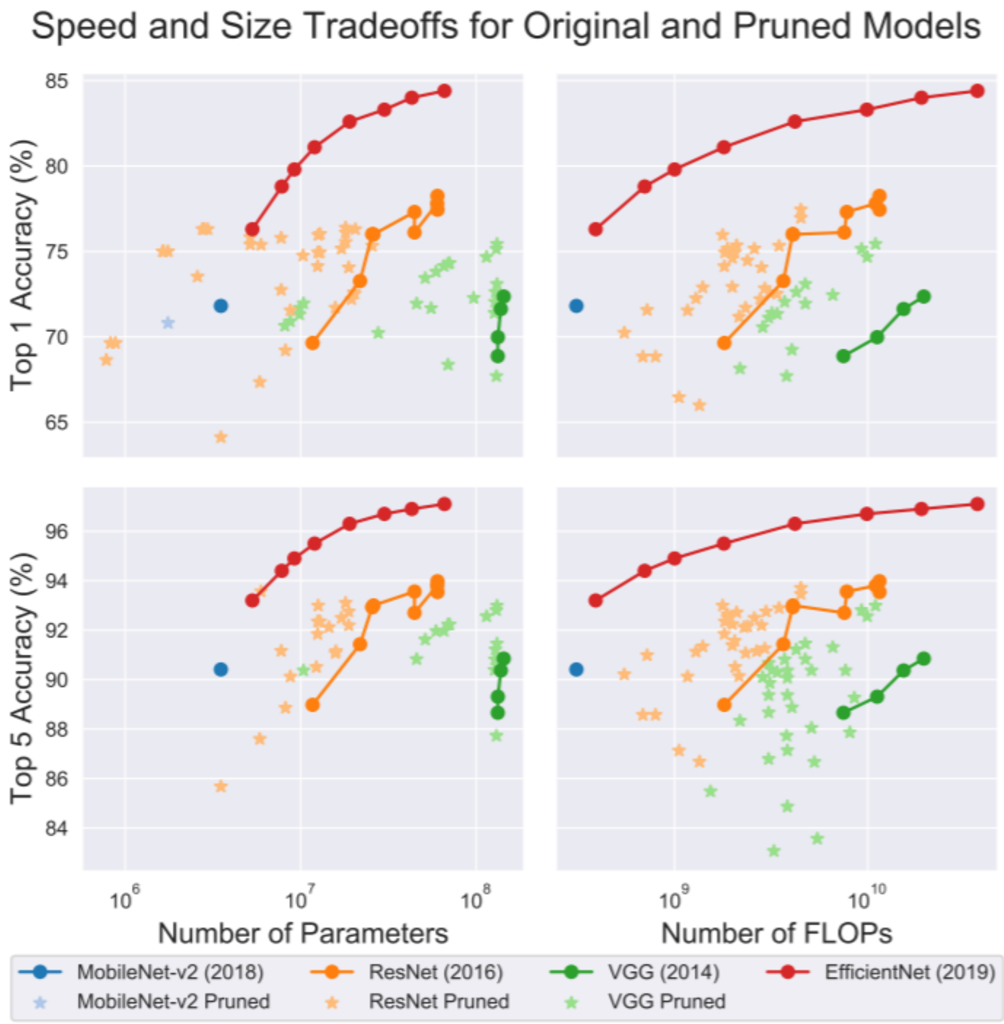


To learn more about pruning: Derrick Mwiti, *Research Guide: Pruning Techniques for Neural Networks*, November 2019

Compression Techniques: Pruning <cont>

Pruning has a few potential **drawbacks:**

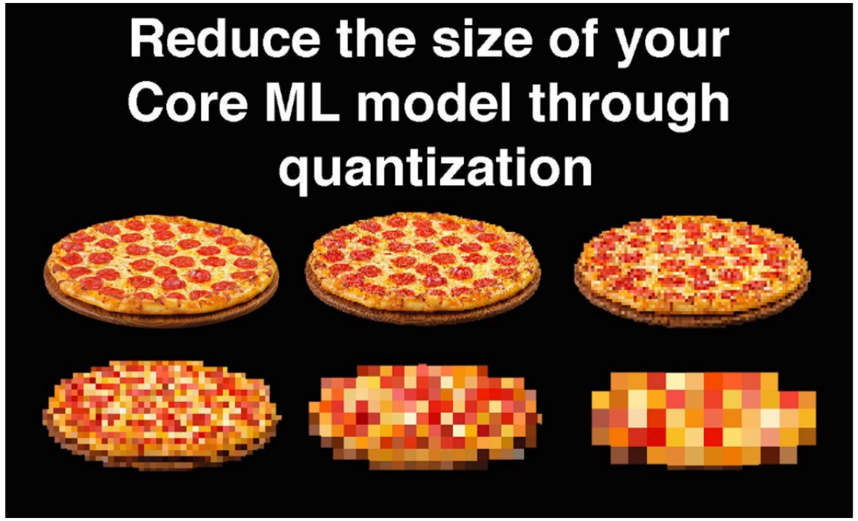
- **Unclear how well given methods generalize** across different architectures.
- **Fine-tuning is cumbersome** and can slow down implementation.
- May be more effective to simply use a **more efficient architecture than to prune a suboptimal one.**



Blalock D. et al, *What is the state of neural network pruning?*, March 2020

Compression Techniques: Quantization

Main idea is to map values from a large set to values in a smaller set without losing too much information in the process. So by reducing the number of pixels, the image below should still be clear.

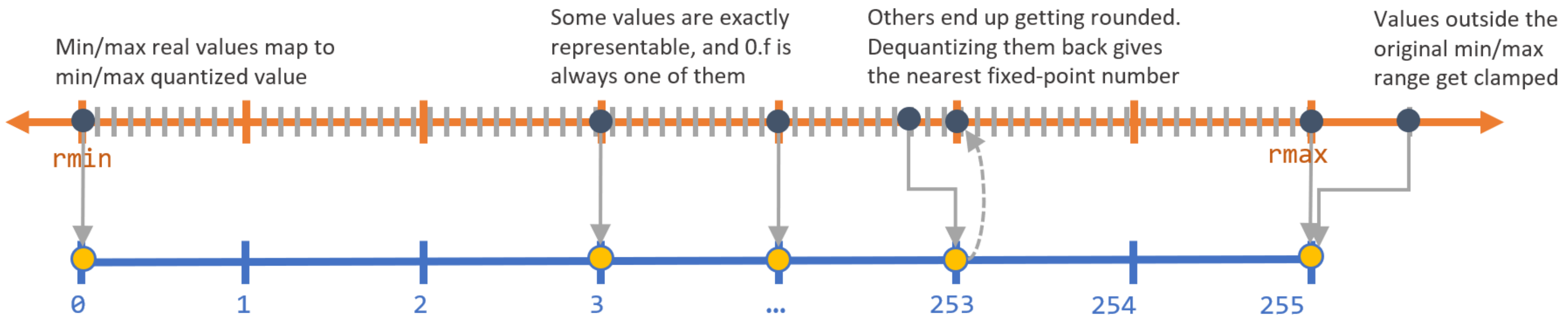


To implement quantization with Tensorflow: MC.AI, [Quantization in Deep Learning using TensorFlow 2.X](), May 2020

Compression Techniques: Quantization

Quantization can be achieved by changing the output or NN architecture:

- **Post Training Quantization:** reducing the size of the weights stored (e.g. from 32-bit floating point numbers to 8-bit)



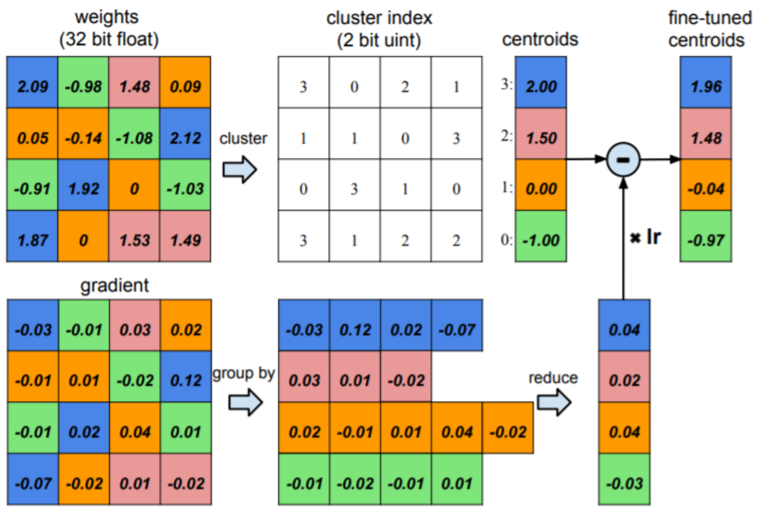
To implement quantization with Tensorflow: MC.AI, *Quantization in Deep Learning using TensorFlow 2.X*, May 2020

Compression Techniques: Quantization <cont>

Quantization-Aware Training:

There could be an accuracy loss in a post-training model quantization and to avoid this and if you don't want to compromise the model accuracy we do quantization aware training.

This technique ensures that the forward pass matches precision for both training and inference.



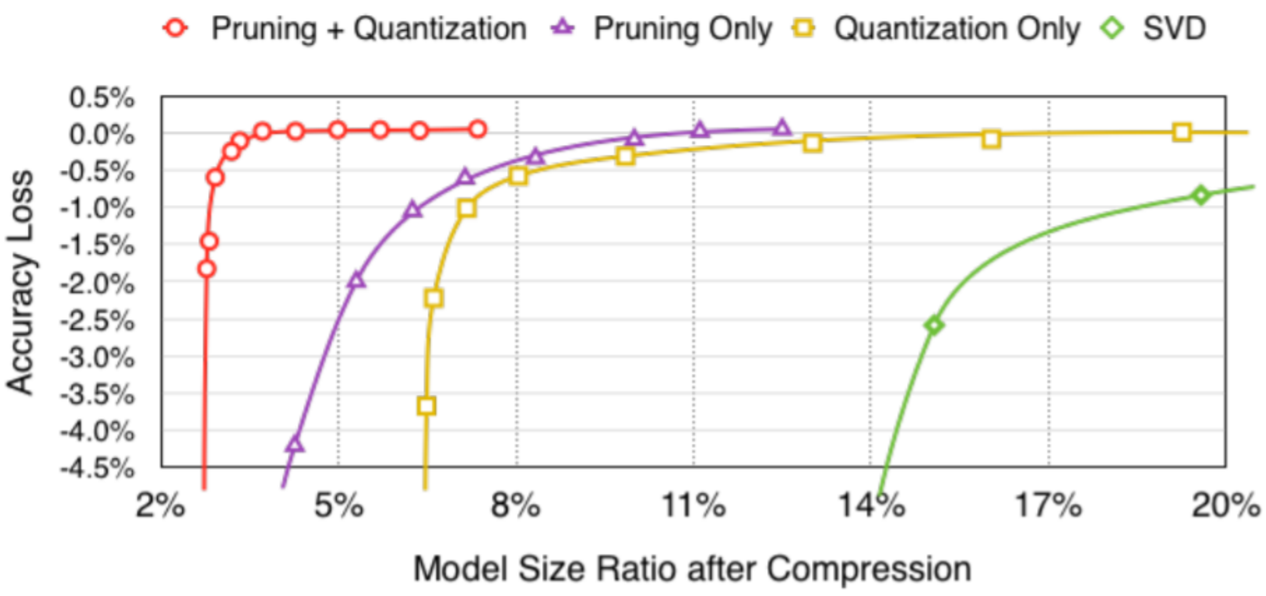
https://www.tensorflow.org/model_optimization/guide/quantization/training

Han S. et al, [Deep compression: compressing deep neural networks with pruning, trained quantization and huffman coding](), 2016

Compression Techniques: Quantization <cont>

Quantization can be **tricky:**

- Requires having a decent **understanding of hardware and bit-wise computations**
- **Savings are tied to the features of the hardware** being used



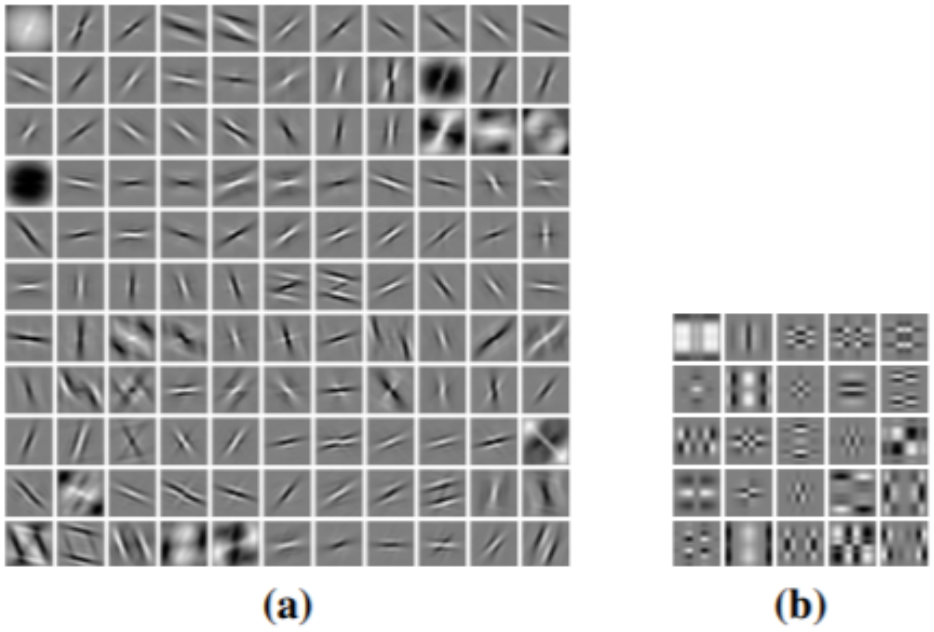
Han S. et al, [*Deep compression: compressing deep neural networks with pruning, trained quantization and huffman coding*](), 2016

Low Rank Approximantion

Main idea is to **approximate the redundant filters of a layer** using a linear combination of fewer filters. Compressing layers in this way reduces the network's memory footprint, the computational complexity of convolutional operations and can yield significant **speedups**.

Examples:

- Singular Value Decomposition
- Tucker decomposition
- Canonical Polyadic decomposition



Rigamonti R. et al., *Learning Separable Filters*, 2013

Low Rank Approximantion <cont>

Kim et al. use Tucker decomposition to determine the ranks that the compressed layers should have. They apply the compression to various models for image classification tasks and run them on both a Titan X and Samsung Galaxy S6 phone*:

- Low-rank approximation achieve significant size and latency reductions
- Prove potential deployment on mobile devices
- Reduce parameters simplifying model structure
- Does not require specialized hardware to implement

Model	Top-5	Weights	FLOPs	S6		Titan X
AlexNet	80.03	61M	725M	117ms	245mJ	0.54ms
*AlexNet**	78.33	11M	272M	43ms	72mJ	0.30ms
(imp.)	(-1.70)	(×5.46)	(×2.67)	(×2.72)	(×3.41)	(×1.81)
VGG-S	84.60	103M	2640M	357ms	825mJ	1.86ms
*VGG-S**	84.05	14M	549M	97ms	193mJ	0.92ms
(imp.)	(-0.55)	(×7.40)	(×4.80)	(×3.68)	(×4.26)	(×2.01)
GoogLeNet	88.90	6.9M	1566M	273ms	473mJ	1.83ms
*GoogLeNet**	88.66	4.7M	760M	192ms	296mJ	1.48ms
(imp.)	(-0.24)	(×1.28)	(×2.06)	(×1.42)	(×1.60)	(×1.23)
VGG-16	89.90	138M	15484M	1926ms	4757mJ	10.67ms
*VGG-16**	89.40	127M	3139M	576ms	1346mJ	4.58ms
(imp.)	(-0.50)	(×1.09)	(×4.93)	(×3.34)	(×3.53)	(×2.33)

* S6 has a GPU with 35× lower computing ability and 13× smaller memory bandwidth than Titan

Kim et al, [Compression of deep convolutional neural networks for fast and low power mobile applications](), 2016

Compression Technique: Distillation

Problem:

- During training, a model does not have to operate in real time and does not necessarily face restrictions on computational resources, as its primary goal is simply to extract as much structure from the given data as possible.
- But latency and resource consumption do become of concern if it is to be deployed for inference.

So what? we must develop ways to compress model for inference.

Compression Technique: Distillation <cont>

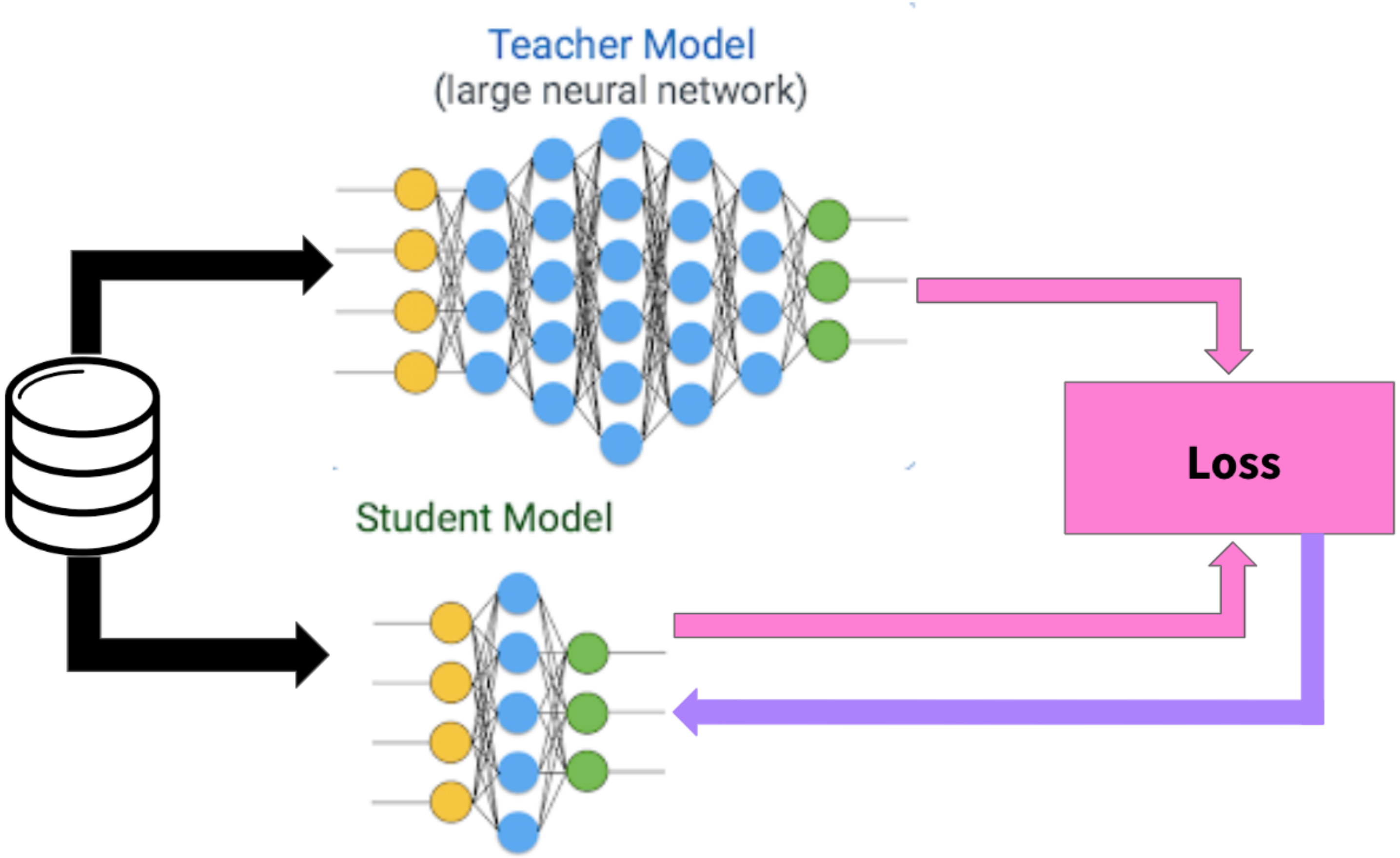
Idea:

- In 2006, Buciluǎ et al. showed that it was possible to transfer knowledge from a large trained model (or ensemble of models) to a smaller model for deployment by **training it to mimic the larger model's output.**
- In 2014 Hinton et al generalized the process and gave the name **Distillation.**

Main idea of distillation is that **training and inference are 2 different tasks;** thus **a different model should be used**.

Buciluǎ et al., *Model Compression*, 2006
Hinton et al., *Distilling the Knowledge in a Neural Network*, 2014

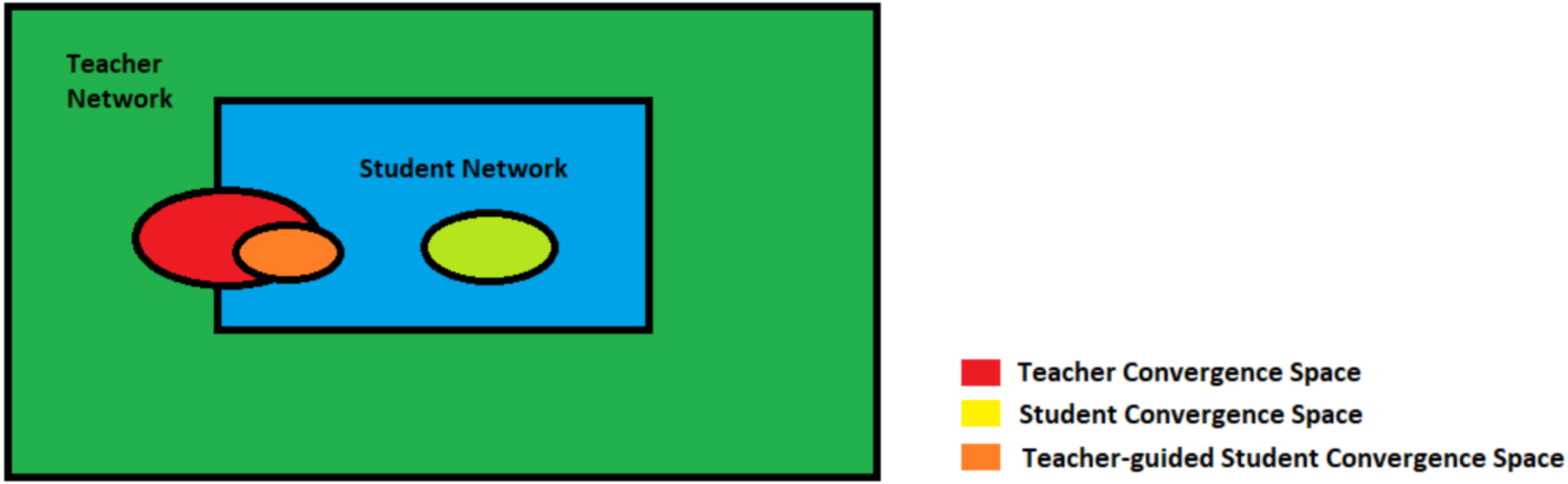
Distillation: Teacher Student <cont>



Source: https://towardsdatascience.com/knowledge-distillation-simplified-dd4973dbc764

Distillation: Teacher Student

Assumption: if we can achieve similar convergence using a smaller network, then the convergence space of the Teacher Network should overlap with the solution space of the Student Network. (design diagram again if needed)



Distillation: Teacher Student Loss <cont>

Modified softmax function with Temperature:

$$qi = \frac{\exp\left(\frac{z_i}{T}\right)}{\sum_j \exp\left(\frac{z_j}{T}\right)}$$

q_i : resulting probability
z_i : logit of a class
z_j : other logits
T: temperature (T=1, “hard output”)

An example of hard and soft targets

cow	dog	cat	car	
0	1	0	0	original hard targets
10^{-6}	.9	.1	10^{-9}	output of geometric ensemble
.05	.3	.2	.005	softened output of ensemble

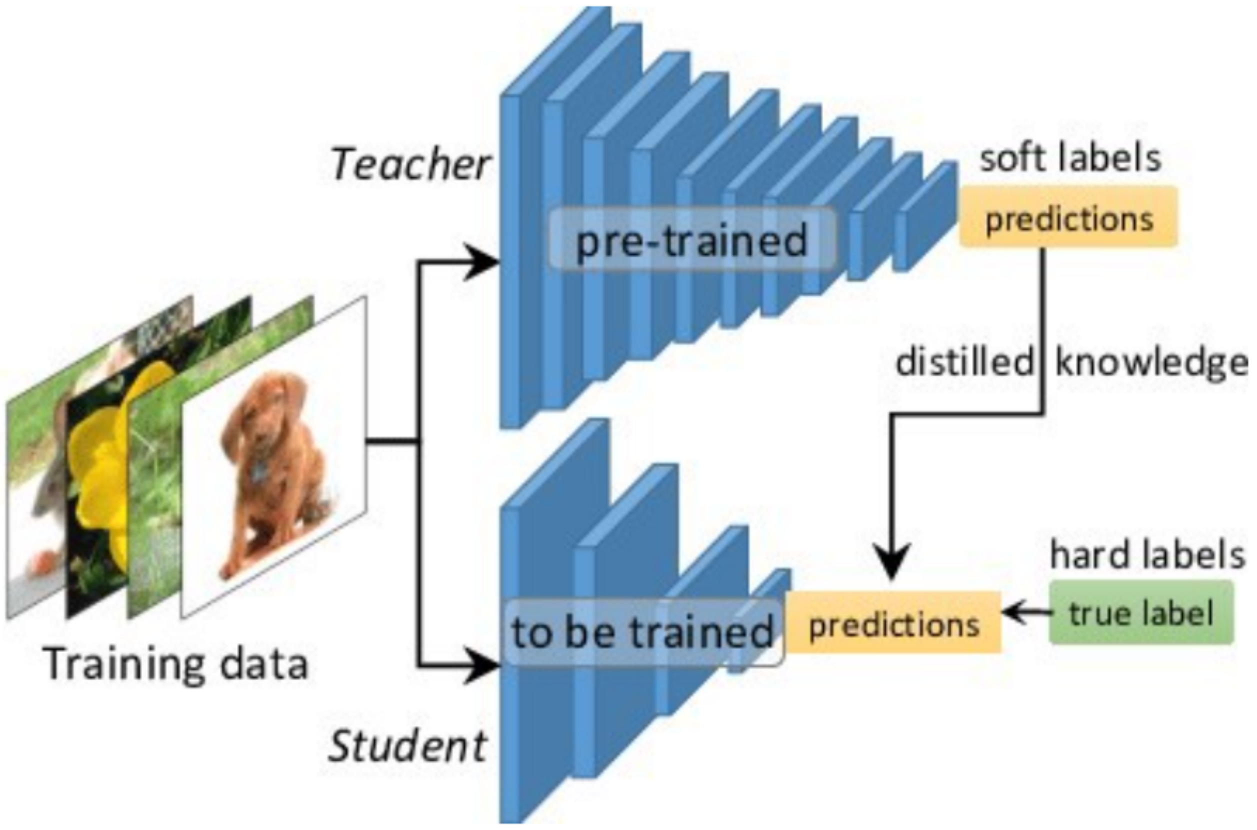
Softened outputs reveal the dark knowledge in the ensemble.

Geoffrey Hinton, Oriol Vinyals & Jeff Dean, *Dark Knowledge*

Distillation: Teacher Student Training <cont>

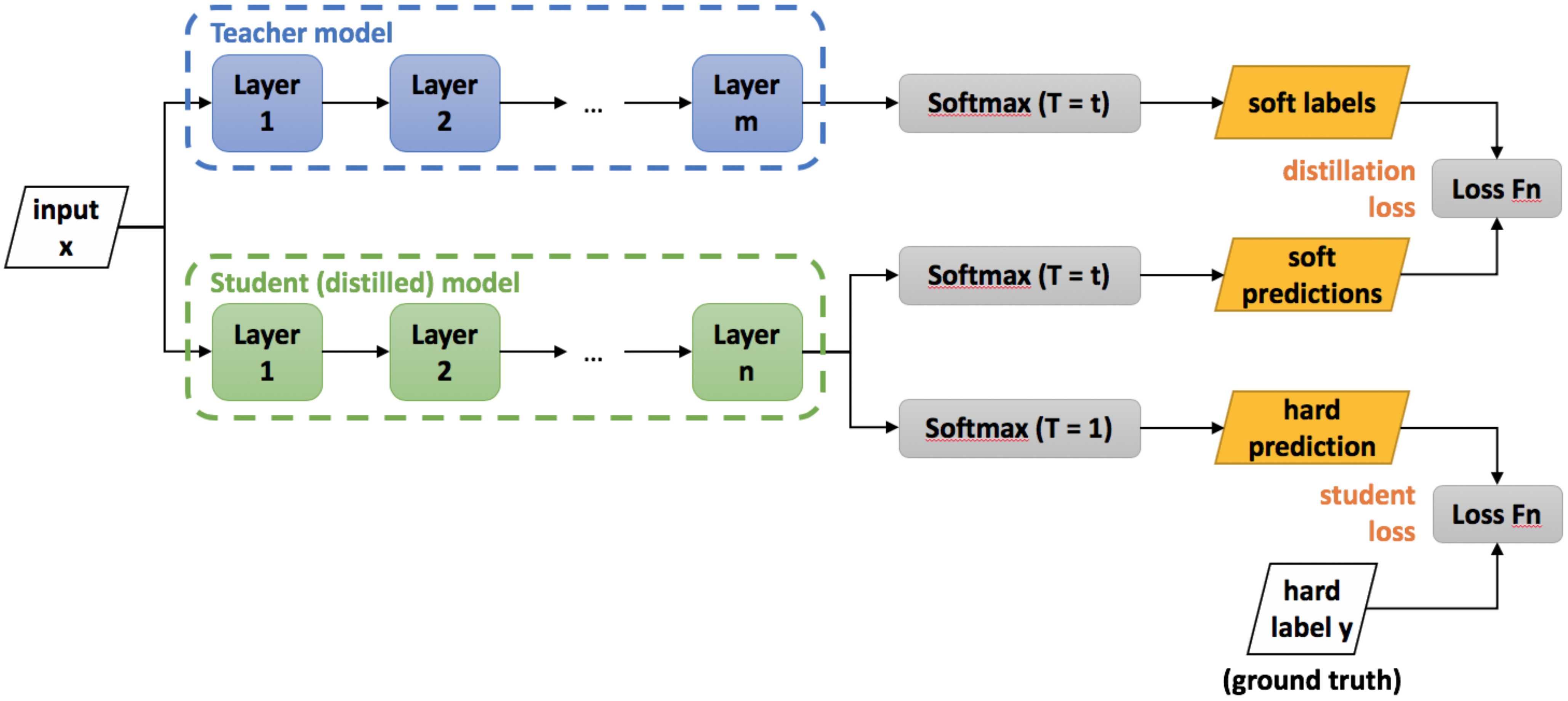
Trained to minimize the sum of two different cross entropy functions:

- one involving the original hard labels obtained using a softmax with T=1
- one involving the softened targets, T>1



Geoffrey Hinton, Oriol Vinyals & Jeff Dean, [Dark Knowledge]

Distillation: Teacher Student Training



Source: https://medium.com/neuralmachine/knowledge-distillation-dc241d7c2322

What is next in Distillation?

1: Multiple teacher (i.e. converting an ensemble into a single network).

2: Introducing a teaching assistant (the teacher first teaches the TA, who then in turn teaches the student) etc.

3: Quite young field

A **drawback** of knowledge distillation as a compression technique, therefore, is that there are many decisions that must be made up-front by the user to implement it (student network doesn't even need to have a similar structure to the teacher).

To the notebook

[LECTURE 9: Compression Techniques and Distillation]

THANK YOU