Lecture 11: Models Compression Techniques

AC215

Pavlos Protopapas
SEAS/Harvard

Announcements / Reminders

- **Milestone 2 Due - 10/18 9PM EST**

Please add Staff GitHub Account - @ac2152024 (or ac215.2024@gmail.com) as collaborator (See Ed post)

Upcoming –

- **Guest Lecture - Modal Labs 10/24**

- **Milestone 3 - Midterm Presentations - 10/31** (Note: For some groups, presentations may begin and end 15-30 minutes later due to room availability.)

Motivation

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 enormous number of images coming from space (8Kx8K pixels from 3K satellites)
3. You cannot run "big" LLMs locally.

Using machine learning is resource intensive:

i. Computing power to train millions/billions of parameters or predict for many observations

ii. Limited bandwidth

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

- Knowledge distillation
- Pruning

Compression Techniques

- **Knowledge distillation**
- Pruning

Compression Technique: Distillation

Compression Technique: Distillation

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

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

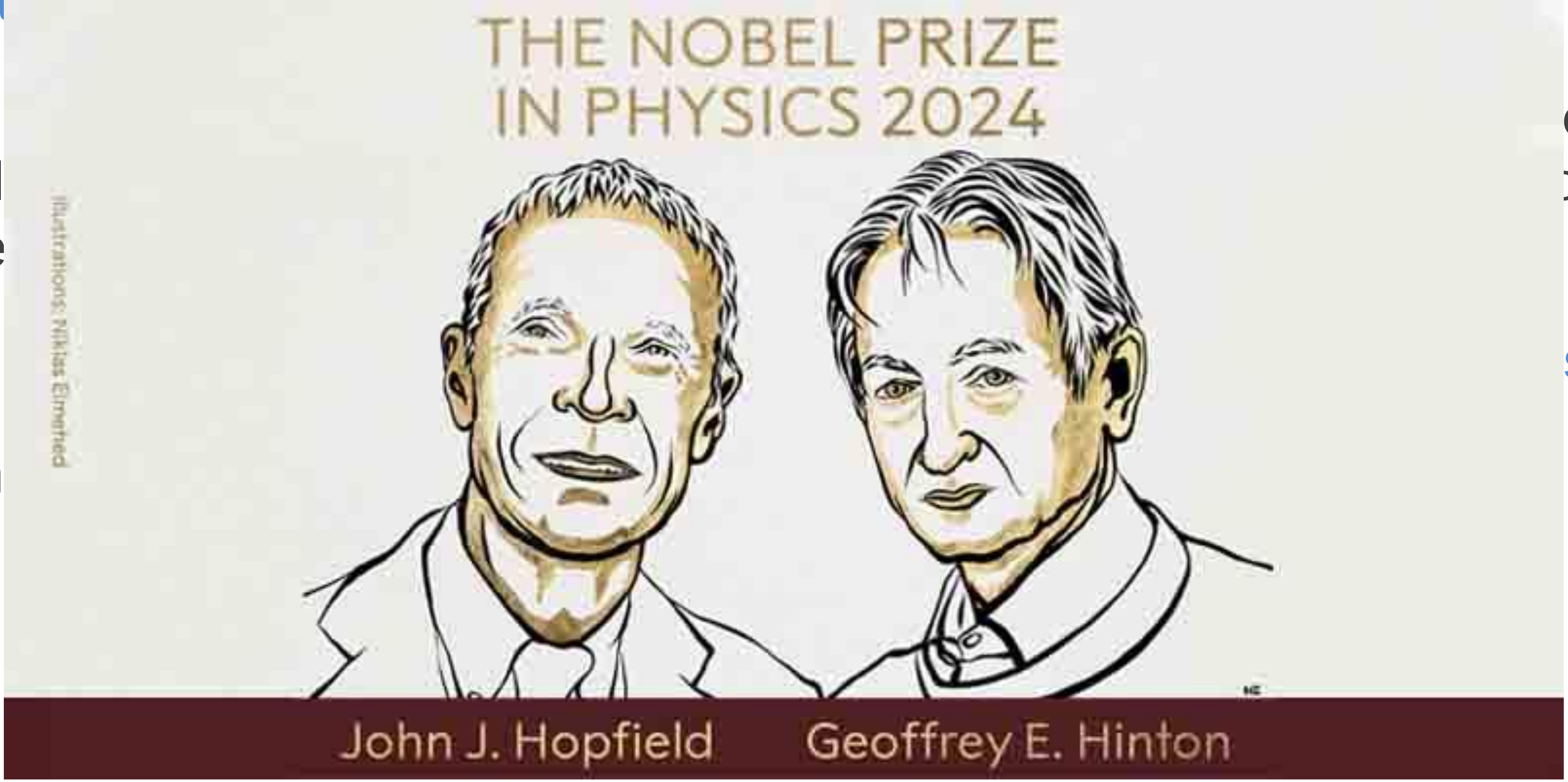
Compression Technique: Distillation

Idea:

- In edge from a l or de

- In **stillation.**

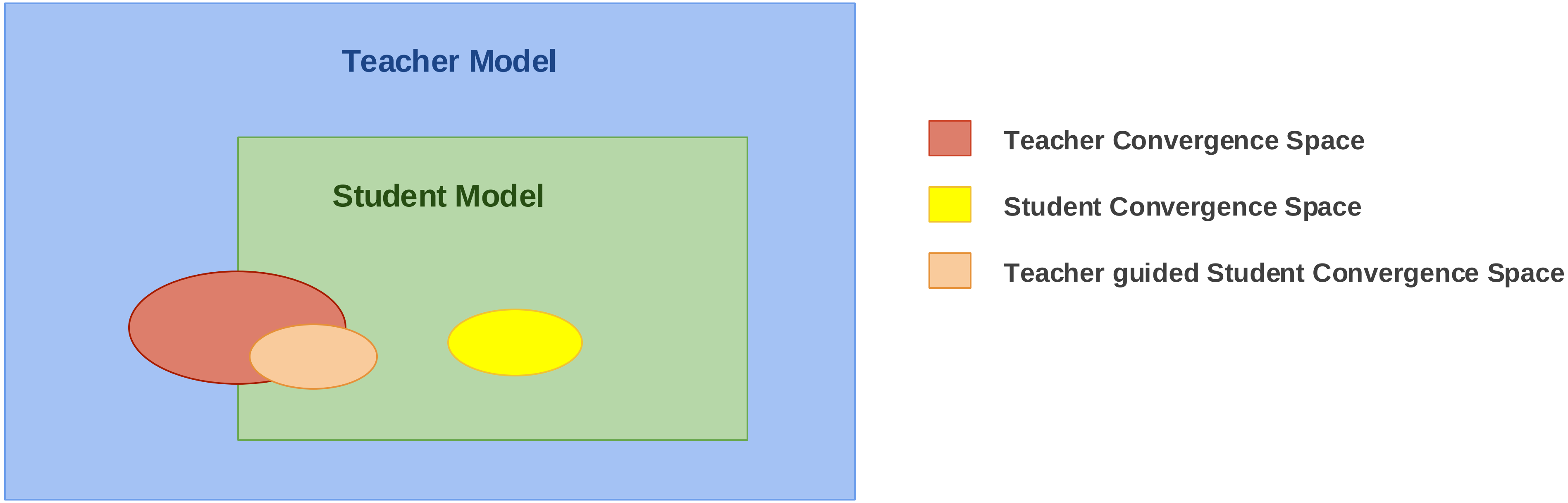
Main **tasks;** thus



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

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.


Teacher Model
Student Model
Teacher Convergence Space
Student Convergence Space
Teacher guided Student Convergence Space

Distillation: Soft targets

Training on one-hot encoded labels only give information about a specific class with complete certainty.

This ignores the rich taxonomy that might exist on a dataset.

Models trained in this setting, will be overconfident in the predictions, hurting the prediction on unseen data.

J. Deng, W. Dong, R. Socher, L. -J. Li, Kai Li and Li Fei-Fei, ImageNet: A large-scale hierarchical image database, 2009

Distillation: Soft targets

While training, our confidence in the labels must be high.

Otherwise the data would not be present in the training set.

For example:

This is a Dog

$$\begin{pmatrix}0\\ \vdots\\ 1\\ 0\\ \vdots\\ 0\end{pmatrix}$$

Distillation: Soft targets

While training, our confidence in the labels must be high.

Otherwise the data would not be present in the training set.

For example:

This is also a dog →

$$\begin{pmatrix}0\\ \vdots\\ 1\\ 0\\ \vdots\\ 0\end{pmatrix}$$

Distillation: Soft targets

While training, our confidence in the labels must be high.

Otherwise the data would not be present in the training set.

For example:

This is also a dog

$$\begin{pmatrix}0\\ \vdots\\ 1\\ 0\\ \vdots\\ 0\end{pmatrix}$$

Distillation: Soft targets

While training, our confidence in the labels must be high.

Otherwise the data would not be present in the training set.

For example:

$$\text{This is a Cat} \longrightarrow \begin{pmatrix}0\\ \vdots\\ 0\\ 1\\ \vdots\\ 0\end{pmatrix}$$

Distillation: Soft targets

While training, our confidence in the labels must be high.

Otherwise the data would not be present in the training set.

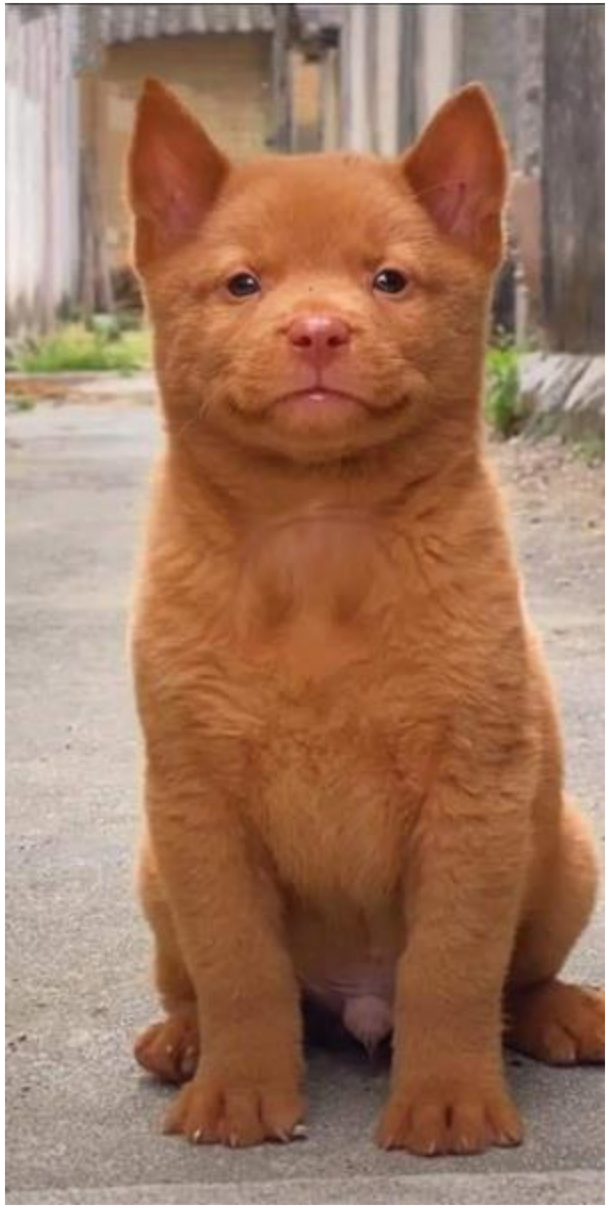
For example:

$$\begin{array}{c}\text{This is also}\\ \text{a Cat}\end{array} \rightarrow \begin{pmatrix}0\\ \vdots\\ 0\\ 1\\ \vdots\\ 0\end{pmatrix}$$

Distillation: Soft targets

What about this example?

This is a … $\begin{pmatrix} 0 \\ \vdots \\ 0.5 \\ 0.5 \\ \vdots \\ 0 \end{pmatrix}$?



Distillation: Soft targets

In the real world, the distinction is never that clear.

Ideally, the training labels must be soft, to learn about the nuanced relations across the elements of the dataset.

Distillation: Teacher - Student

The teacher shows the student the relations between the different classes, based on what it learned during its training stage.

From a dataset containing the original categories, the teaching outputs the normalized probabilities for each example.

cow	car	dog	cat	
0	1	0	0	Original labels
10^{-6}	0.9	0.1	10^{-9}	Probabilities

Adapted from:
Geoffrey Hinton, Oriol Vinyals & Jeff Dean, *Dark Knowledge*

Distillation: Teacher - Student

To enhance the student’s learning, the teacher softens its output *z* even more.

To do this, introduces the softmax function with temperature *T*:

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

cow	car	dog	cat	
0	1	0	0	Original labels
10^{-6}	0.9	0.1	10^{-9}	Softmax probabilities
0.05	0.3	0.2	0.005	Softened probabilities

Adapted from:
Geoffrey Hinton, Oriol Vinyals & Jeff Dean, *Dark Knowledge*

Distillation: Teacher - Student

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

A high value of *T* will dilute the information contained in the probabilities.

- The standard softmax is recovered for *T*=1.
- For T>1, the probabilities are softened. Typical values range between 2 and 5.
- For T<1, it approaches to the original one-hot labels.

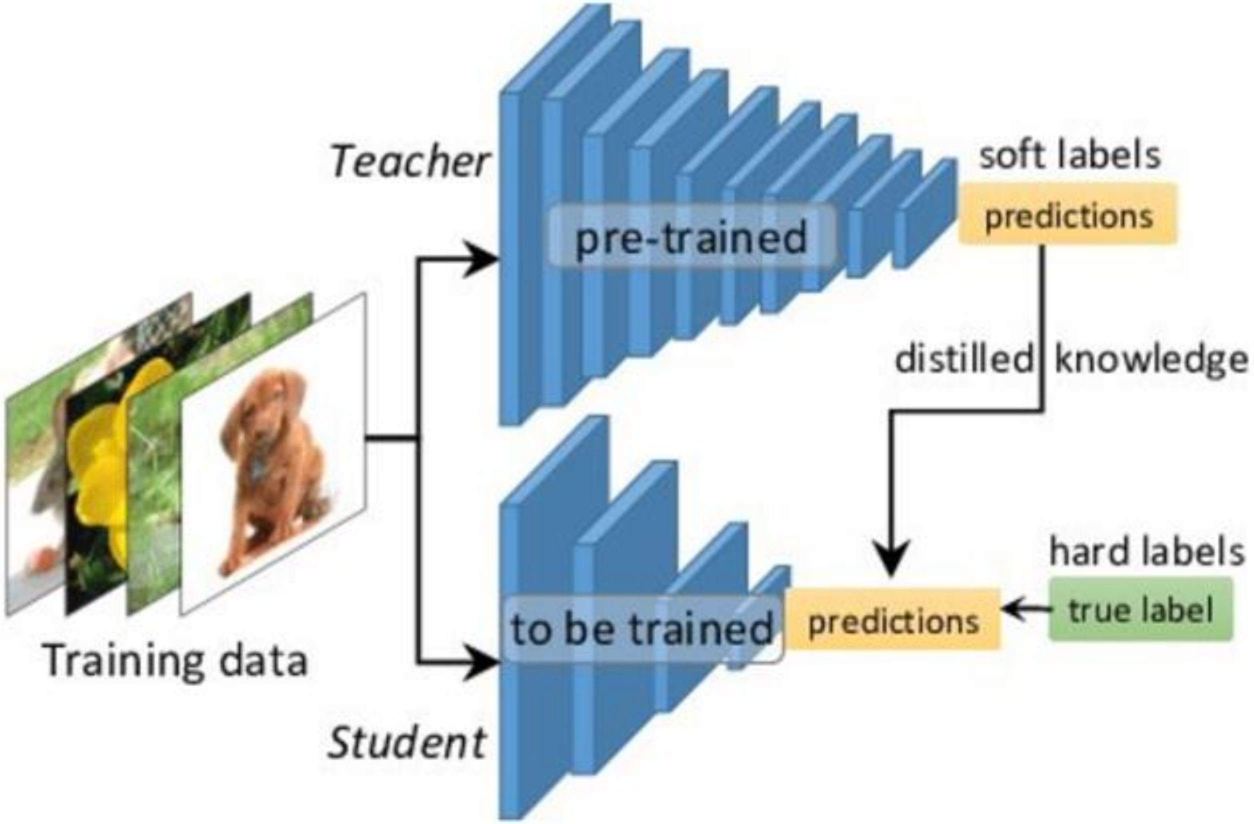
cow	car	dog	cat	
0	1	0	0	Original labels
10^{-6}	0.9	0.1	10^{-9}	Softmax probabilities
0.05	0.3	0.2	0.005	Softened probabilities

Adapted from:
Geoffrey Hinton, Oriol Vinyals & Jeff Dean, *Dark Knowledge*

Temperature explanation

Distillation: Teacher Student Training

The student is trained to solve the original problem, and to replicate the softened probabilities from the teacher.



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

Distillation: Teacher Student Training

Scenario 1:

Train on the same data as the teacher model. DistilBert was trained on this example.

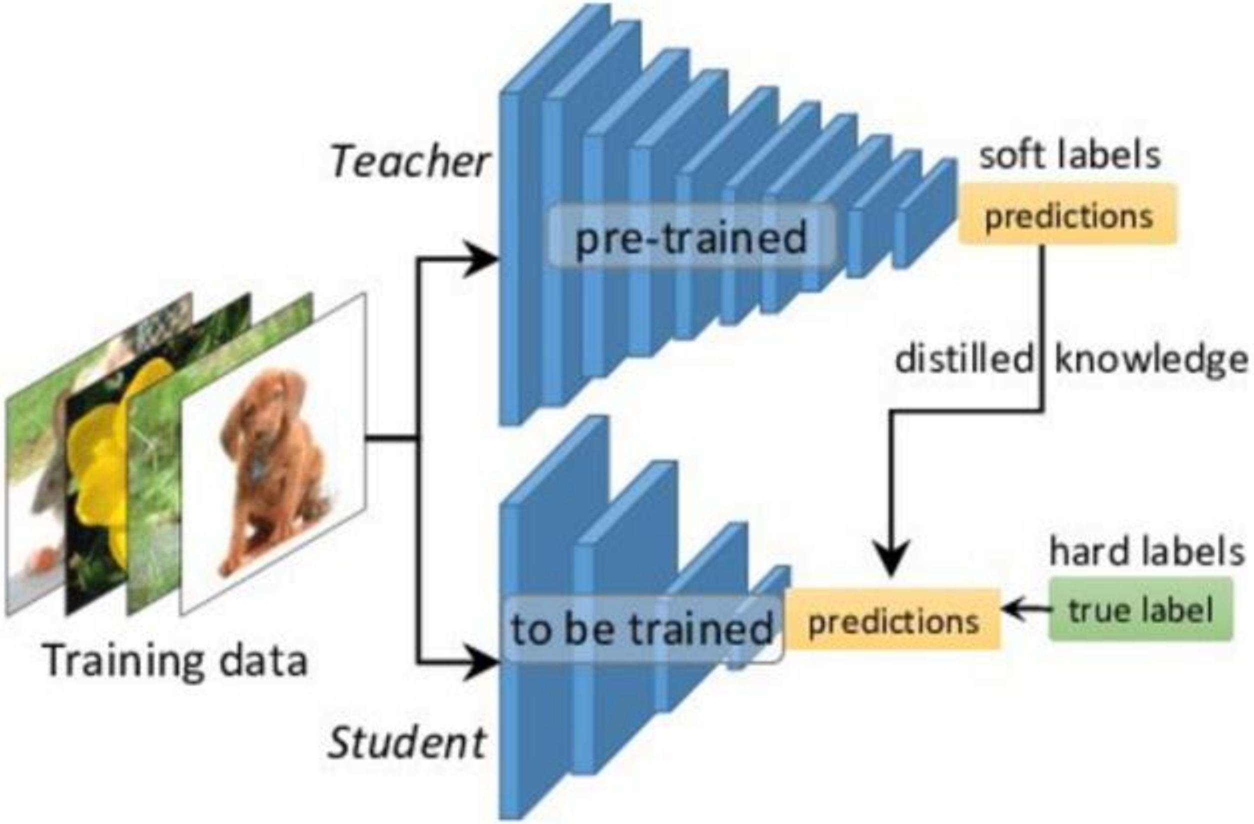
Scenario 2:

Train on a reduce dataset. This is usually done when you have the model but not the original dataset or resources.

Distillation: Teacher Student Training

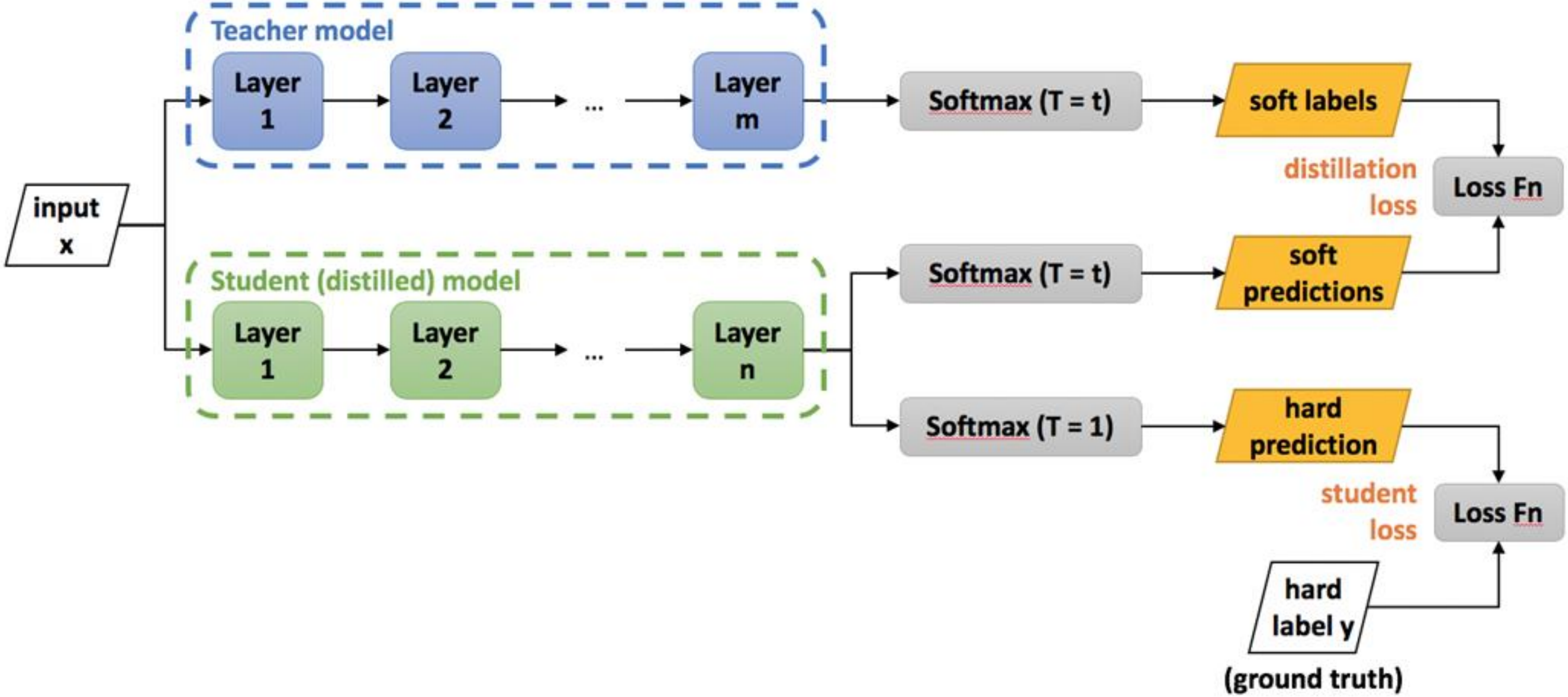
It minimizes the sum of two different cross entropy functions:

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



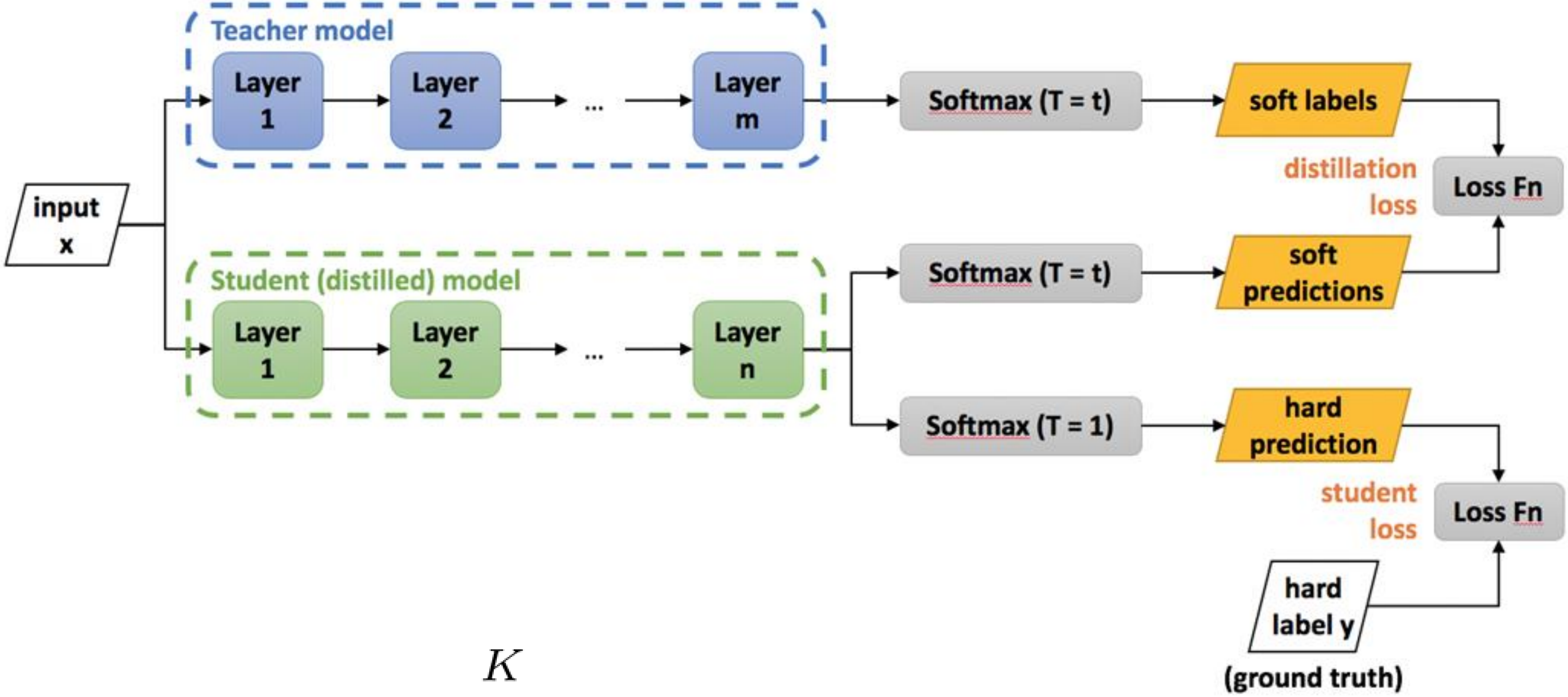
Geoffrey Hinton, Oriol Vinyals & Jeff Dean, *Dark Knowledge*

Distillation: Teacher Student Training



$$L = \lambda L_{student} + (1 - \lambda) L_{distillation}$$

Distillation: Teacher Student Training



$$L_{Distillation} = -\sum_{k}^{K} p_T^k(x,T)\log\left(q_S^k(x,T)\right)$$

Distillation: Teacher Student Training

$$L_{Distillation} = -\sum_{k}^{K} p_T^k(x,T)\log\left(q_S^k(x,T)\right)$$

In our example, for the example *n* in the dataset.

$$p_T(x,T) = [0.05, 0.3, 0.2, 0.005]$$

And our predictions

$$q_T(x,T) = [0.1, 0.45, 0.30, 0.15]$$

The resulting loss value for this example is:

$$-[0.05 \cdot \log(0.1) + 0.3 \cdot \log(0.45) + 0.2 \cdot \log(0.30) + 0.005 \cdot \log(0.15)]$$

Tutorial: Model Distillation

Colab Notebook

What is next in Distillation?

1: Multiple teachers (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: Learning not only the output, but the intermediate representations.

4: Quite a *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).

Compression Techniques

- Knowledge distillation
- **Pruning**

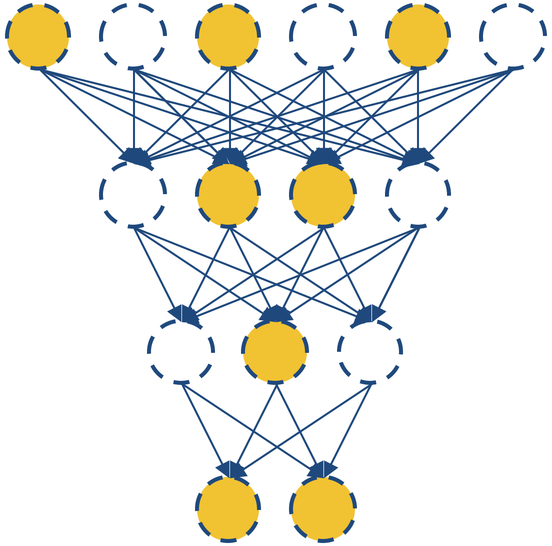
Compression Technique: Pruning

Pruning is based on the idea of the Lottery Ticket Hypothesis.

A randomly-initialized neural network contains a subnetwork that is initialized such that - when trained in isolation- it can match the test accuracy of the original network after training for at most the same number of iterations.

Inside each network, we might find a winning ticket where a small subset of the neurons are non-zero.

In principle, this network will be use less memory, perform faster during inference without losing performance.



[The lottery ticket hypothesis: Finding sparse, trainable neural networks](#)
[Frankle, J., & Carbin, M. (2018)]

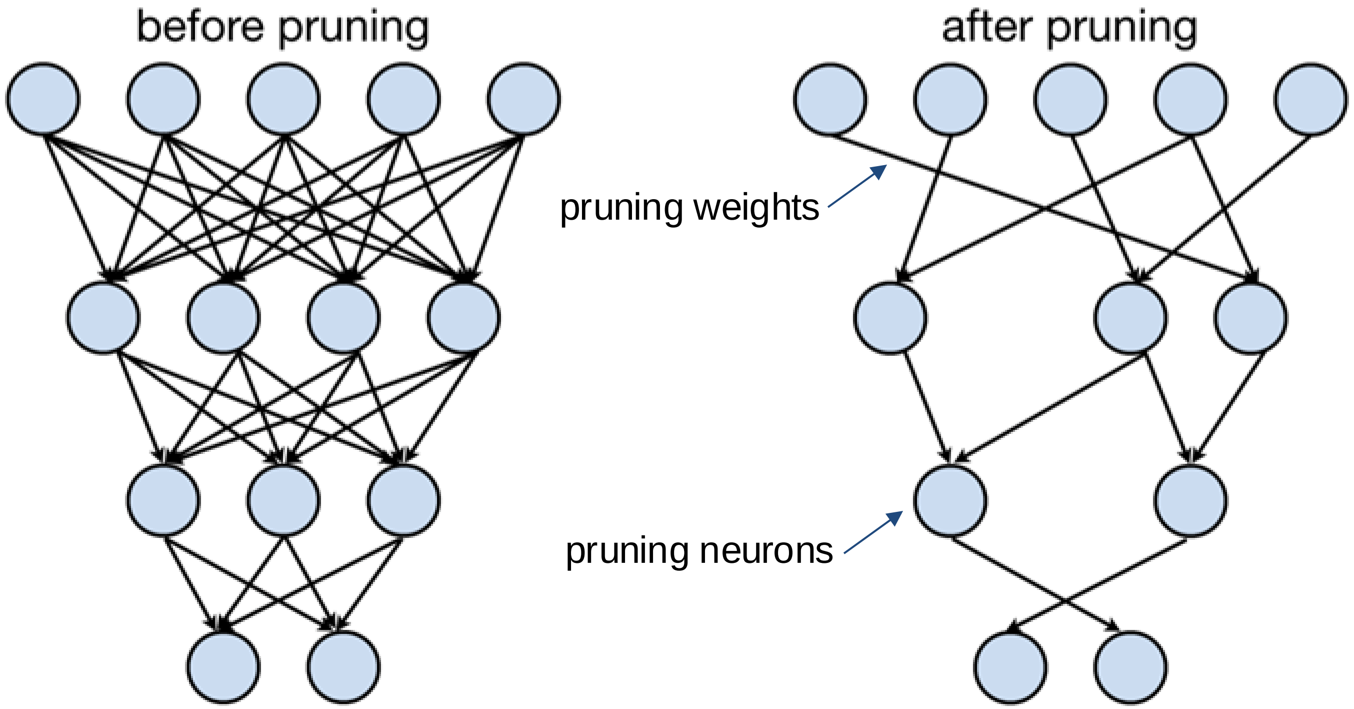
Compression Technique: Pruning

The main idea is to remove less impactful features of the neural network.

Pruning involves removing connections between neurons, and neurons from a trained network. To prune a connection, we set a weight in the matrix to zero.

Two types of pruning:

- Pruning weights
- Pruning structure

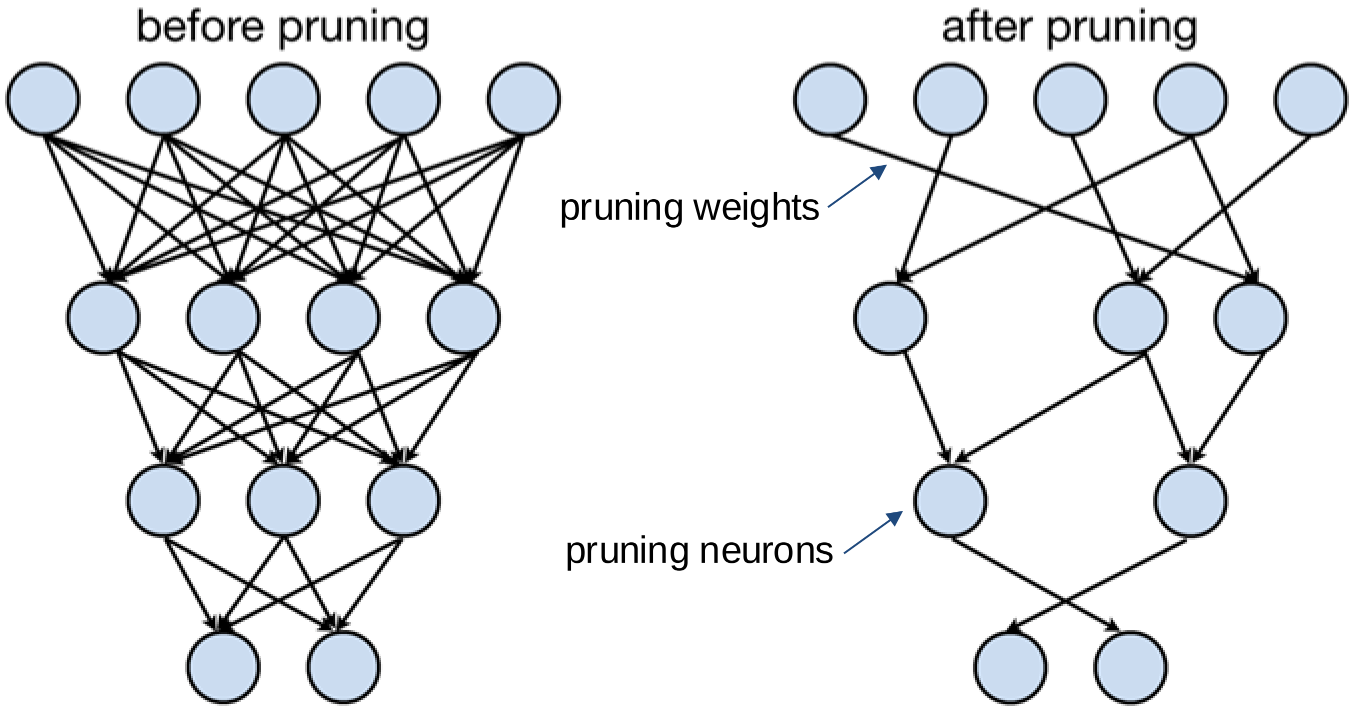


Learning Both Weights and Connections for Efficient Neural Network [Han et al., NeurIPS 2015]

Compression Technique: Pruning

The main idea is to remove less impactful features of the neural network.

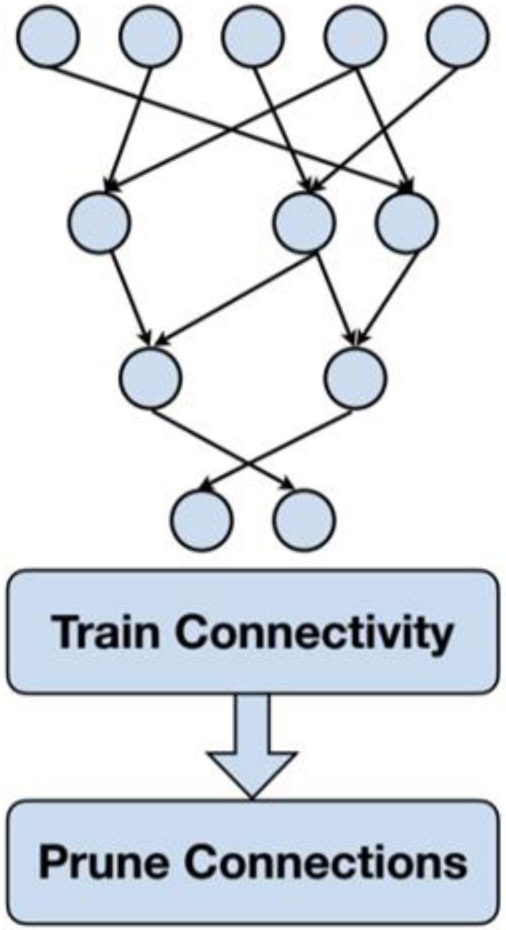
The criteria to select the rank the relevant features might depend on the weights absolute value, on the activations magnitudes, among other values.

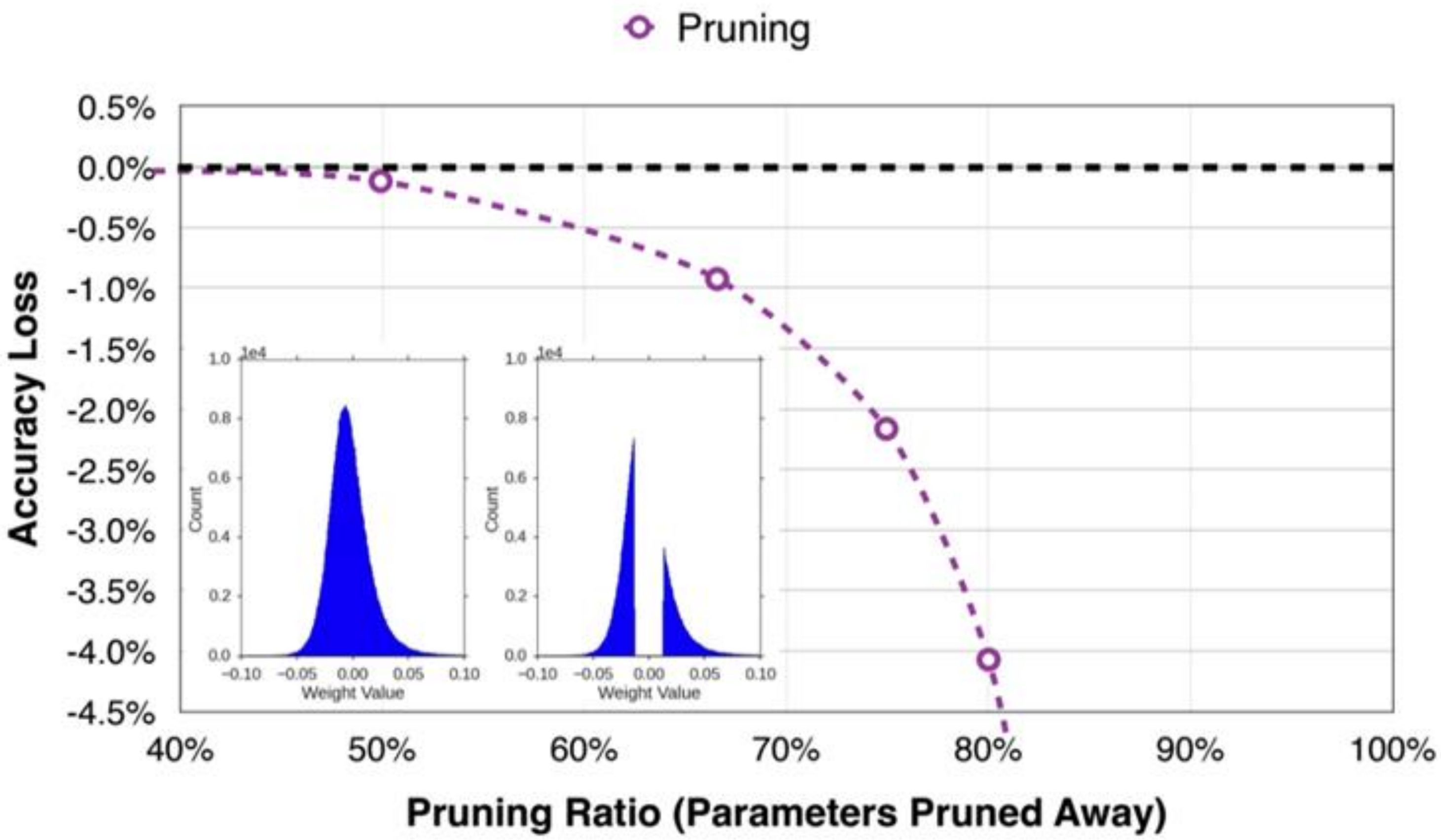


Learning Both Weights and Connections for Efficient Neural Network [Han et al., NeurIPS 2015]

Compression Technique: Pruning

Make neural networks smaller by removing weights and neurons



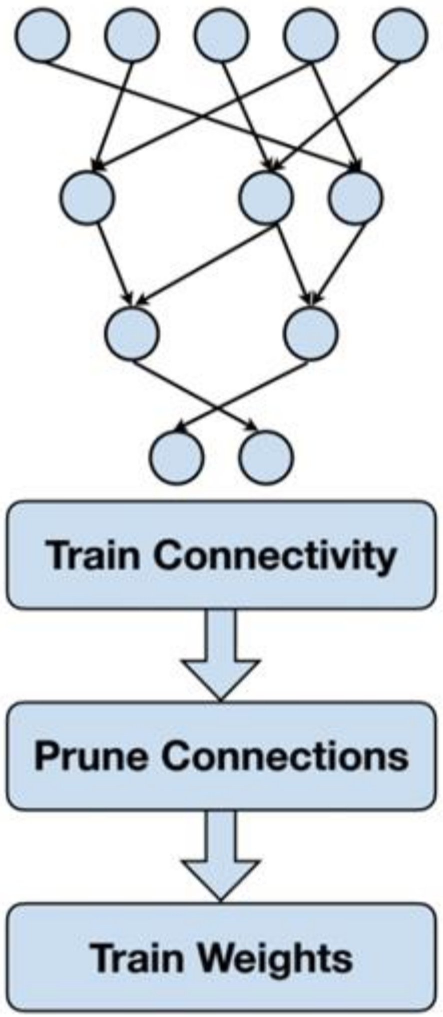


Learning Both Weights and Connections for Efficient Neural Network [Han et al., NeurIPS 2015]

MIT EfficientML.ai: Pruning and Sparsity

Compression Technique: Pruning

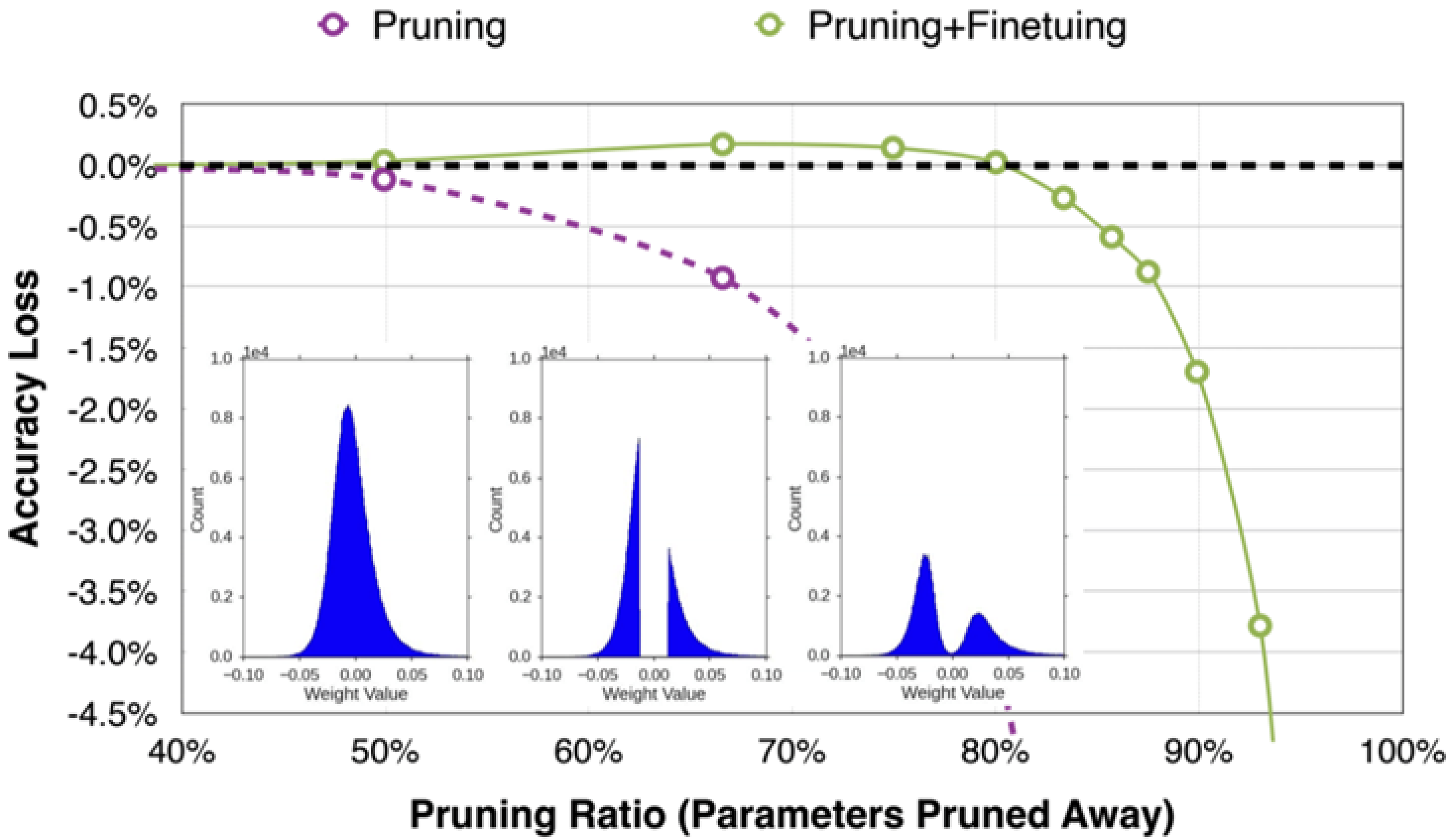
Make neural networks smaller by removing weights and neurons



Train Connectivity

Prune Connections

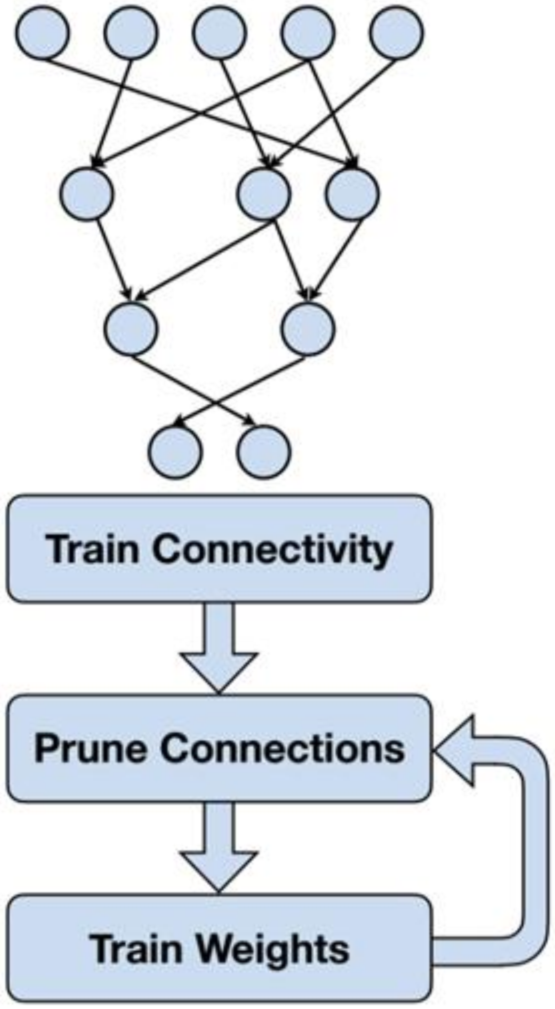
Train Weights



Learning Both Weights and Connections for Efficient Neural Network [Han et al., NeurIPS 2015]
MIT EfficientML.ai: Pruning and Sparsity

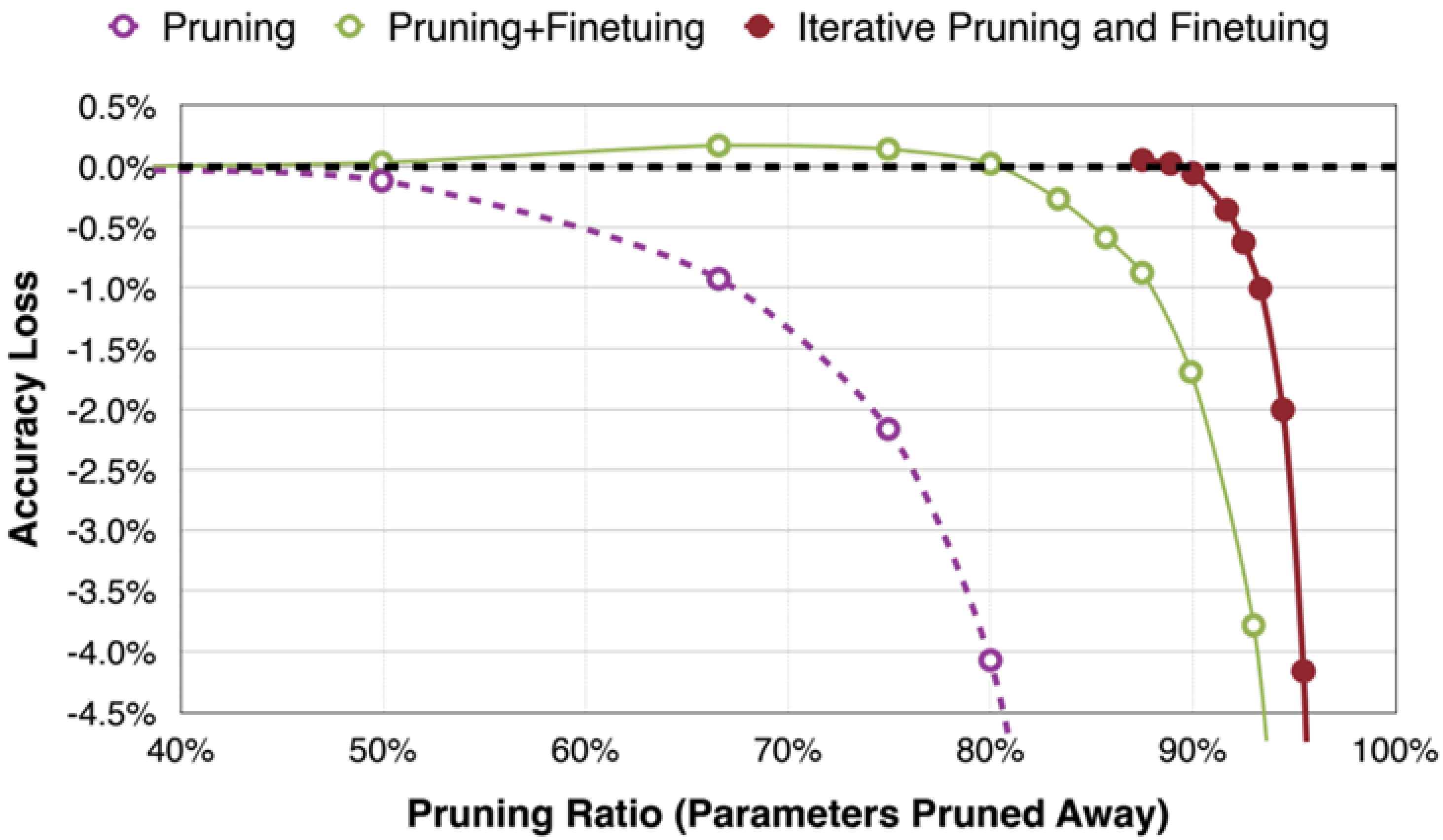
Compression Technique: Pruning

Make neural networks smaller by removing weights and neurons


Train Connectivity

Prune Connections

Train Weights

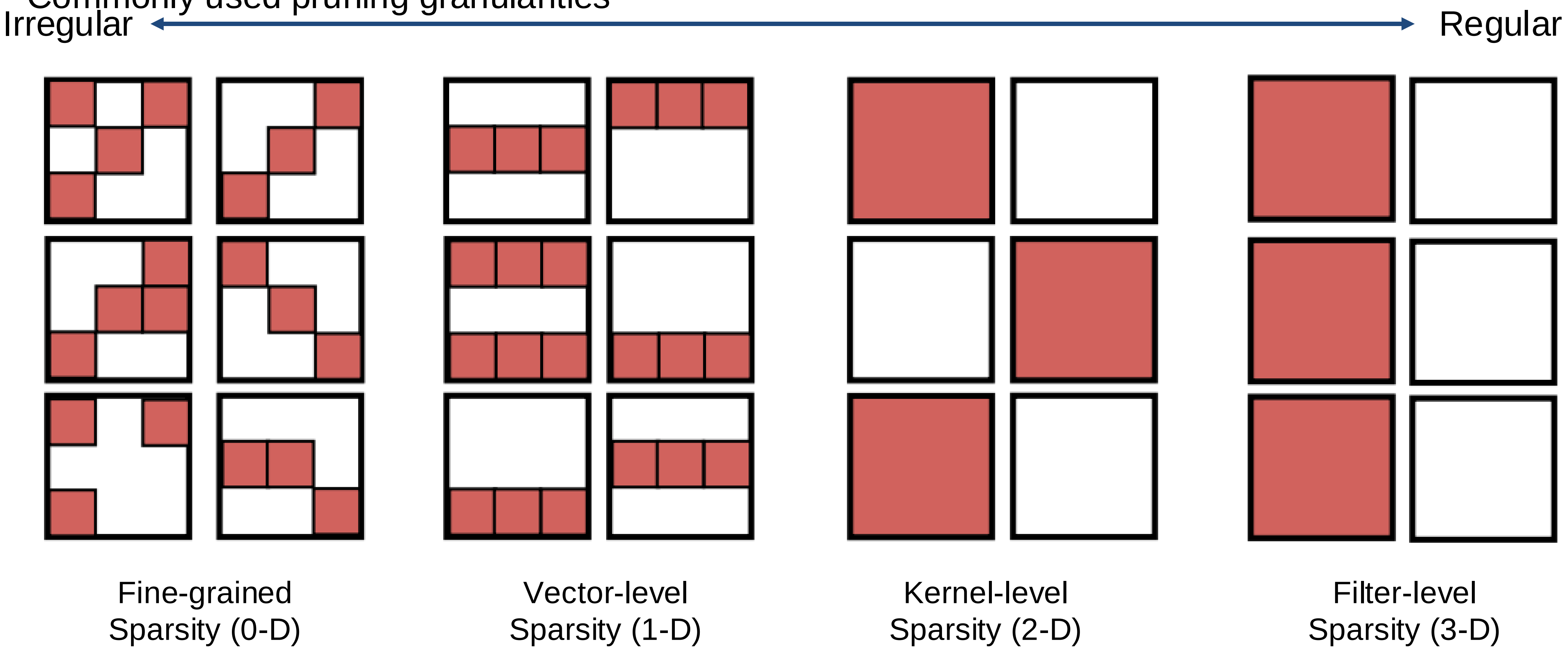


Learning Both Weights and Connections for Efficient Neural Network [Han et al., NeurIPS 2015]
MIT EfficientML.ai: Pruning and Sparsity

Compression Technique: Pruning

Example using convolutional layers

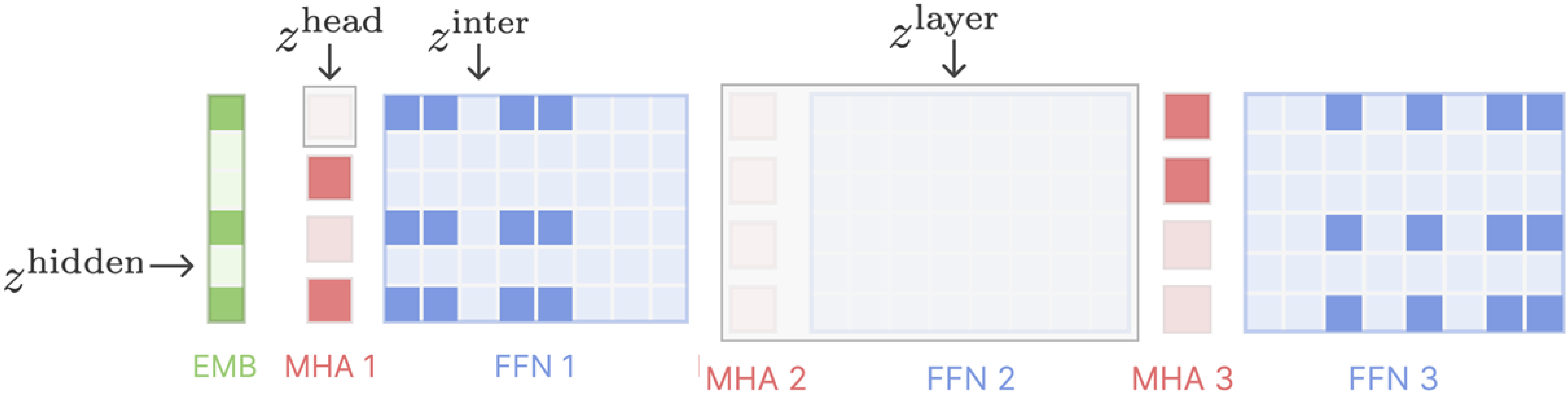
- Commonly used pruning granularities



Exploring the Regularity of Sparse Structure in Convolutional Neural Networks
[Mao et al., CVPR-W]

Compression Technique: Pruning

Pruning on transformers can be applied at different levels



Sheared LLaMA: Accelerating Language Model Pre-training via Structured Pruning [Xia et al.,2024]

Compression Technique: Pruning

Drawbacks of neural network pruning:

- **Optimal Pruning Challenge**: determining the optimal neurons or weights to prune without detrimentally impacting model performance can be complex and time consuming in practise.
- **Fine-Tuning Requirement**: after pruning, models typically require additional fine-tuning to recover potential losses in predictive accuracy, which might consume additional training resources and time.
- **Hardware Dependency**: pruned models might not always translate to proportional computational or energy savings due to hardware inefficiencies or dependencies in exploiting sparsity.
- **Model Robustness**: excessive or imprecise pruning may lead to a substantial decrease in model accuracy or robustness, especially when encountering unseen or out-of-distribution data.

Compression Technique: Next steps

Things to consider:

- Quantization, distillation and pruning applied alone can reduce the models complexity.
- But together, these techniques achieve state-of-the-art performance while keeping a “*reduced size”.*
- Distilled LLMs can still be huge, with ~7 billion parameters.
- The choice of methodology depends on the dataset, tasks and models.
- Finetuning is always necessary to regain performance losses. LoRA is extremely relevant.
- There is no general methodology that will work out of the box. Experiment!

THANK YOU