



TinyML Workshop

Efficient Machine Learning Inference and Deployment.

Obed Mogaka | obed.mogaka@mdu.se

HERO Lab, Malardalens University

November 4, 2025

Overview

1. Introduction to Pruning

- What is pruning?
- How to formulate pruning.

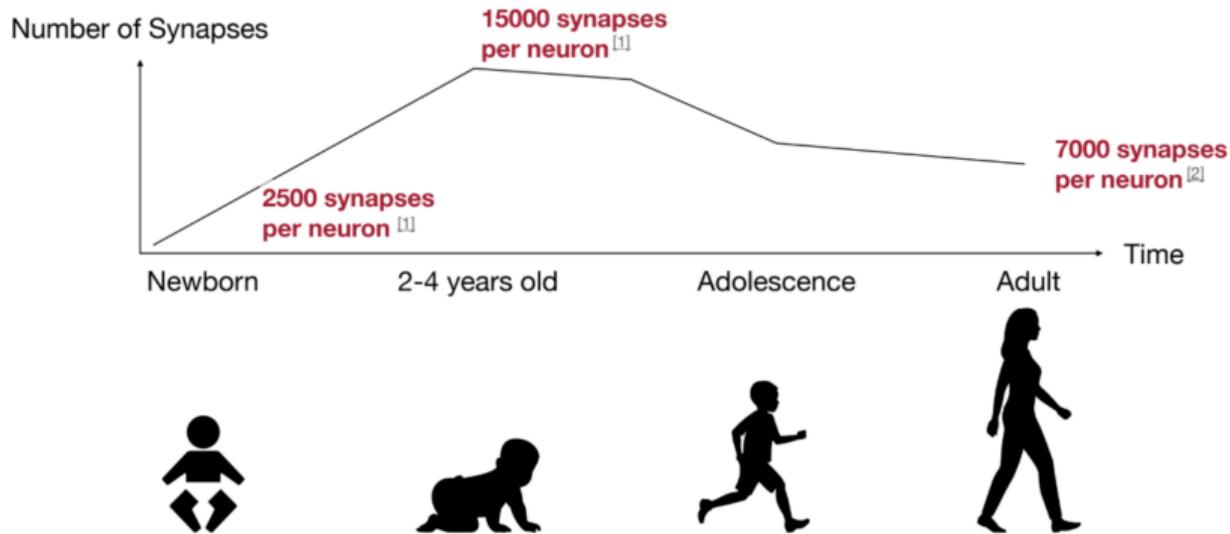
2. How and when to prune?

- Pruning Granularity.
- Pruning Criterion.
- Pruning Ratio.
- Train/Fine-tune

3. Hands-on

Introduction to Pruning

Pruning in the Human Brain



Sources: ^{1,2}, Alila Medical Media

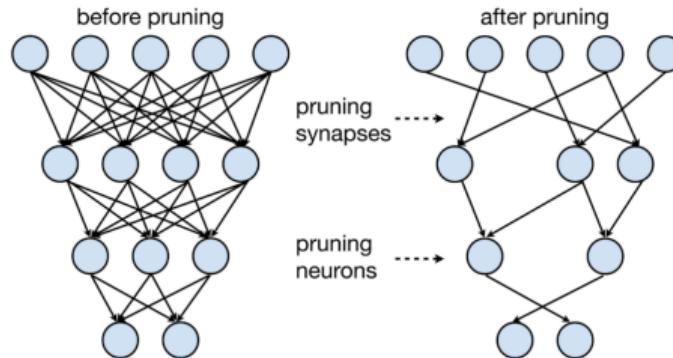
¹Drachman, David A. "Do we have brain to spare?" *Neurology* 64.12 (2005): 2004-2005.

²Walsh, Christopher A. "Peter Huttenlocher (1931–2013)." *Nature* 502.7470 (2013): 172-172.

Neural Network Pruning

Pruning makes a neural network smaller by removing synapses and neurons³

It removes unnecessary/redundant parameters from the model while maintaining predictive performance.



Weight matrix (before pruning)

0.01	0.02	-1.9	0.02	1.76	0.01	0.01	3.75	0.02	0.01	0.01
7.93	0.02	0.01	0.68	0.02	0.01	-1.1	0.01	0.02	0.01	0.01
0.02	0.01	5.2	0.2	0.02	0.01	0.01	0.02	-6.2	0.02	0.01
0.01	0.02	0.01	0.01	0.01	0.01	0.02	0.01	-2.5	0.01	0.01
0.32	0.01	-3.5	0.01	0.88	0.01	0.02	0.01	0.02	0.01	0.02
0.01	0.02	0.02	2.4	0.02	-3.1	0.01	0.01	0.02	0.01	8.36
0.96	9.77	0.92	0.01	0.01	0.01	8.5	6.6	0.01	0.01	0.01
0.03	0.8	0.01	0.03	0.01	0.02	0.03	0.02	0.02	0.01	0.02
0.01	0.02	0.01	0.02	0.01	0.01	0.03	0.7	14.8	0.01	0.91
0.02	0.02	0.01	0.01	0.02	0.01	-0.38	0.01	0.01	0.03	10.1
0.01	0.03	16.3	0.03	0.01	2.9	0.01	0.01	0.02	-5.4	0.01

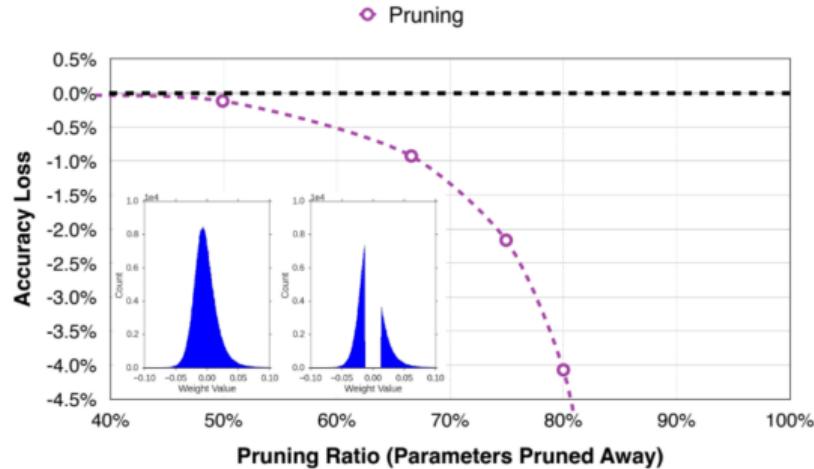
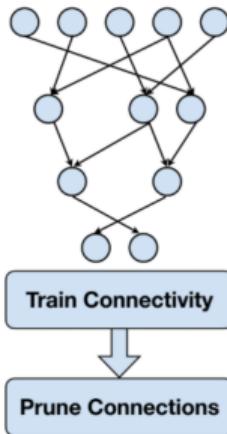
Weight matrix (after pruning – very sparse)

0	0	-1.9	0	1.76	0	0	3.75	0	0	0
7.93	0	0	0.68	0	0	-1.1	0	0	0	0
0	0	5.2	0.2	0	0	0	0	-6.2	0	0
0	0	0	0	0	0	0	0	-2.5	0	0
0.32	0	-3.5	0	0.88	0	0	0	0	0	0
0	0	0	2.4	0	-3.1	0	0	0	0	8.26
0.96	9.77	0.92	0	0	0	8.5	6.6	0	0	0
0	0.8	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0.7	14.8	0	0.91
0	0	0	0	0	0	-0.38	0	0	0	10.1
0	0	16.3	0	0	2.9	0	0	0	-5.4	0

³Han, Song, et al. "Learning both weights and connections for efficient neural network." Advances in neural information processing systems 28 (2015).

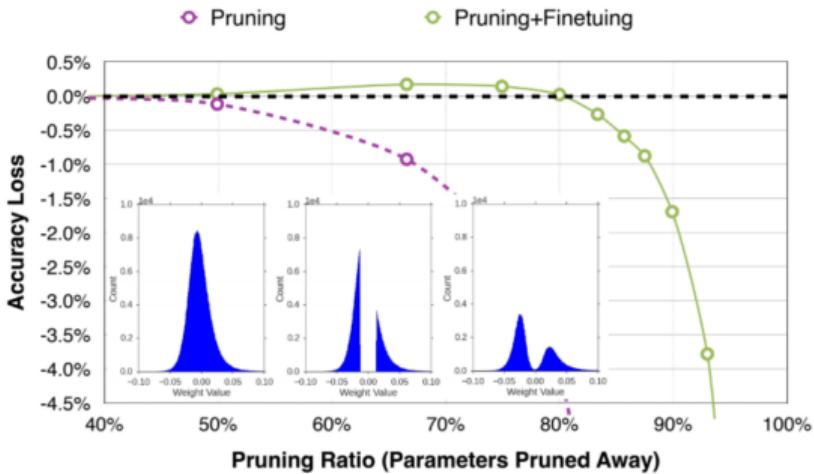
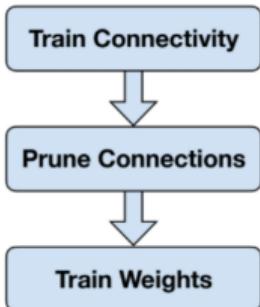
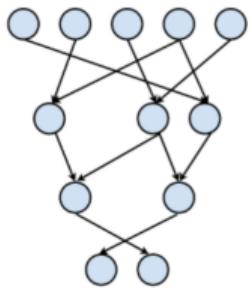
Neural Network Pruning

- When we train a NN, we are learning the connections.
- To make the NN smaller and efficient we prune redundant connections



Aggressive pruning leads to accuracy loss!

Pruning with Fine-tuning



Effect of Pruning

Table 1: Effect of pruning: parameter counts and MACs for common CNNs. Reductions are shown as multiples relative to the unpruned models.⁴

Neural Network	#Parameters		MACs	
	Before Pruning	After Pruning	Reduction	Reduction
AlexNet	61 M	6.7 M	9 ×	3 ×
VGG-16	138 M	10.3 M	12 ×	5 ×
GoogleNet	7 M	2.0 M	3.5 ×	5 ×
ResNet50	26 M	7.47 M	3.4 ×	6.3 ×
SqueezeNet	1 M	0.38 M	3.2 ×	3.5 ×

⁴Han, Song. Efficient methods and hardware for deep learning. Diss. Stanford University, 2017.

Formulating Pruning

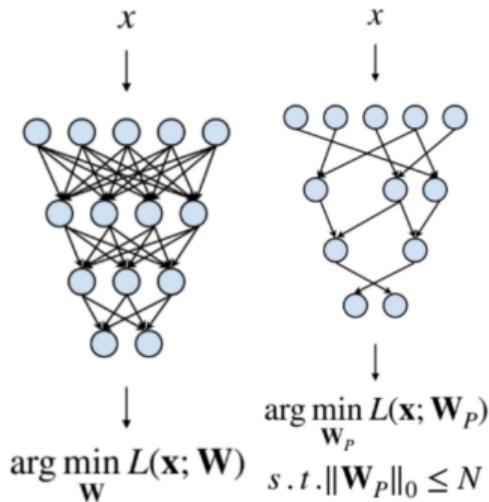
Pruning can be formalized as an optimization problem.

Given a trained model with parameters W , we seek a sparse version W_P that retains only the most important parameters.

- The objective is defined as follows:

$$\arg \min_{W_P} L(\mathbf{x}; \mathbf{W}_P) \text{ subject to } \|\mathbf{W}_P\|_0 < N$$

- L represents the objective function for neural network training;
- \mathbf{x} is the input, \mathbf{W} is the original weights, and \mathbf{W}_P is the pruned weights;
- $\|\mathbf{W}_P\|_0$ calculates the number of nonzero elements in \mathbf{W}_P , and N is the target number of nonzeros.

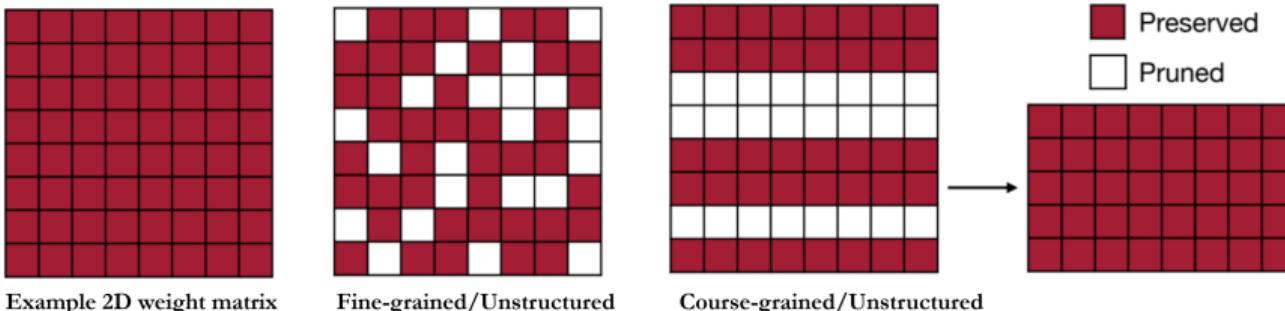


How to Prune

Pruning Granularity

In what pattern should we prune the neural network?

Pruning can be performed at different granularities, from structured to non-structured.



Fine-grained/Unstructured

- More flexible pruning index choice
- Hard to accelerate (irregular)

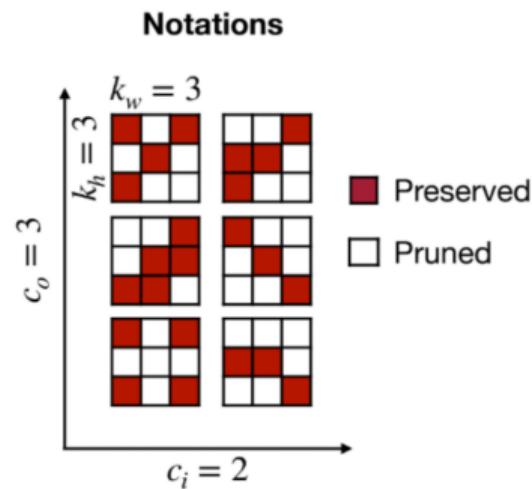
Course-grained/Structured

- Less flexible pruning index choice (a subset of the fine-grained case)
- Easy to accelerate (just a smaller matrix!)

Pruning at Different Granularities

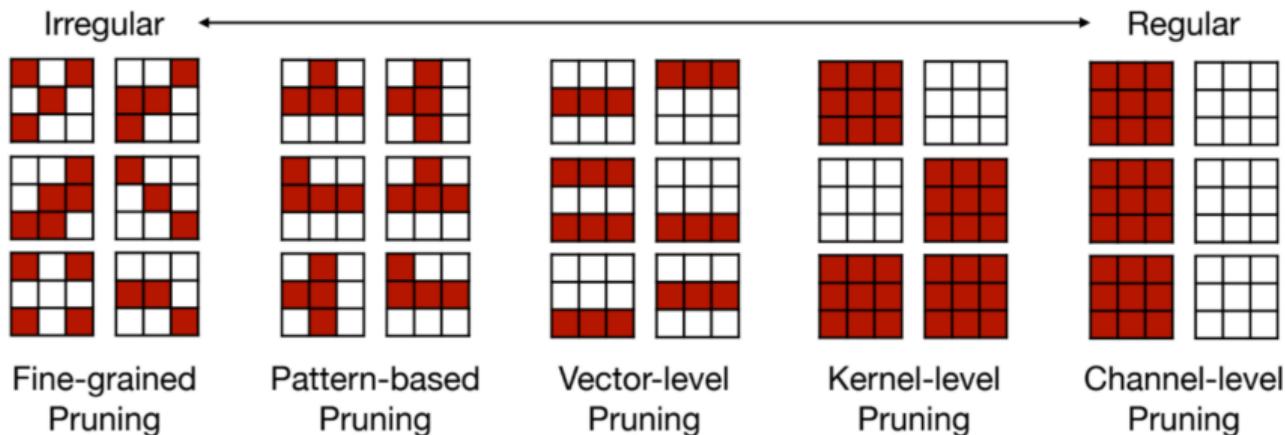
The case of convolutional layers

- The weights of convolutional layers have 4 dimensions $[c_o, c_i, k_h, k_w]$:
 - c_i : input channels (or channels)
 - c_o : output channels (or filters)
 - k_h : kernel size height
 - k_w : kernel size weight
- The 4 dimensions give us more choices to select pruning granularities.



Pruning at Different Granularities

Some of the commonly used pruning granularities

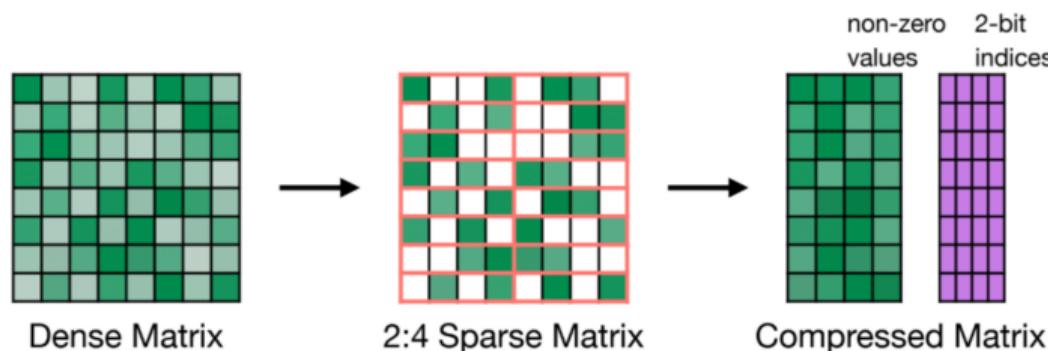


Pattern-based Pruning

Looking into some cases

- **Pattern-based Pruning: N:M sparsity**

- N:M sparsity means that in each contiguous M elements, N of them is pruned.
- A classic case is 2:4 sparsity (50% sparsity)
- Supported by NVIDIA's Ampere GPU Architecture, which delivers up to 2x speed up

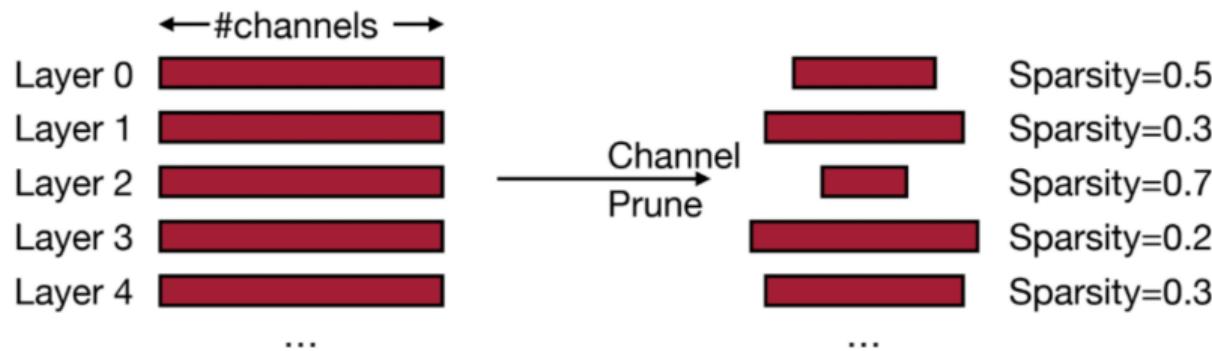


Channel Pruning

Looking into some cases

- **Channel Pruning**

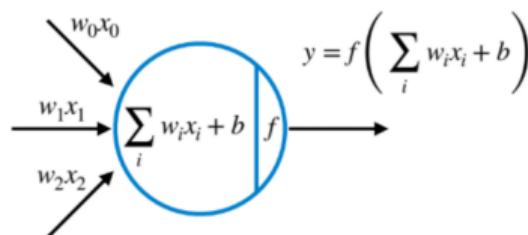
- Pro: Direct speed up due to reduced channel numbers (leading to an NN with fewer channels)
- Con: smaller compression ratio



Pruning Criterion

What synapses and neurons should we prune?

- When removing parameters from a neural network model,
 - the less important the parameters being removed are,
 - the better the performance of pruned neural network is.



Example

$$f(\cdot) = \text{ReLU}(\cdot), \quad W = [10, -8, 0.1]$$

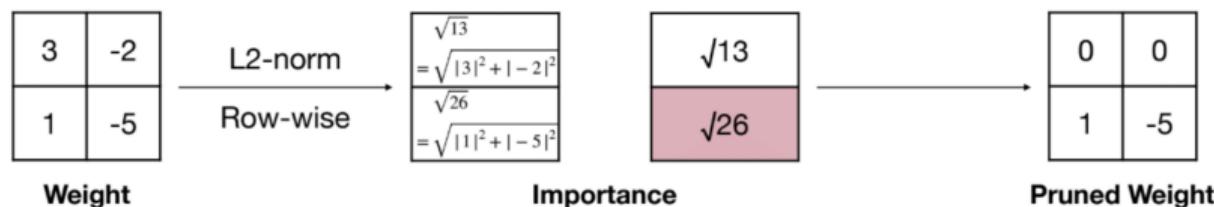
$$\Rightarrow y = \text{ReLU}(10x_0 - 8x_1 + 0.1x_2)$$

- If one weight will be removed, which one?

Magnitude-based Pruning

A heuristic pruning criterion

- Magnitude-based pruning considers weights with **larger absolute values** are more important than other weights.
- Magnitude is also known as $L_p - \text{norm}$ defined as,
$$\|W^{(S)}\|_p = (\sum_{i \in S} |w_i|^p)$$
, where $W^{(S)}$ is a structural set of parameters



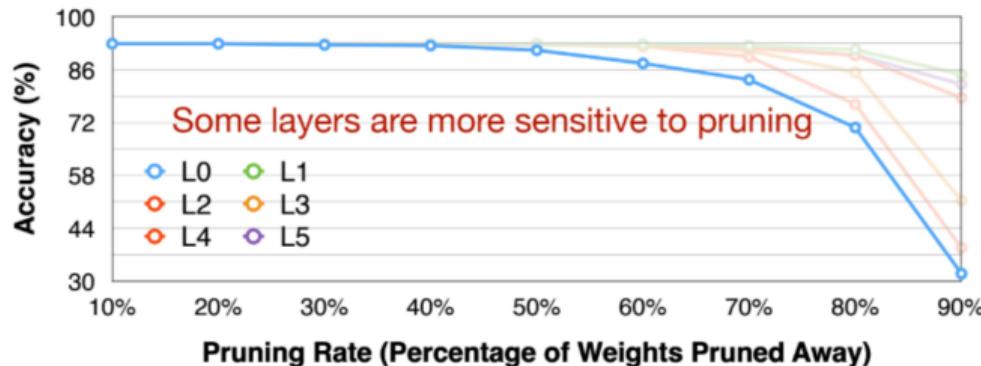
Other Pruning Criteria

- **Filter pruning using scaling factors**
- **Second-Order-based pruning:** Minimize the error on loss function introduced by pruning synapses.
- **Percentage-of-Zero-based pruning:** A measure of the importance of the neurons.
- **Regression-based Pruning:** Minimize reconstruction error of the corresponding layer's outputs.
- **One-shot Pruning:** removes multiple architectural components in a single step, followed by an extensive fine-tuning phase to recover model accuracy.

Finding Pruning Ratios

Analyse the sensitivity of each layer

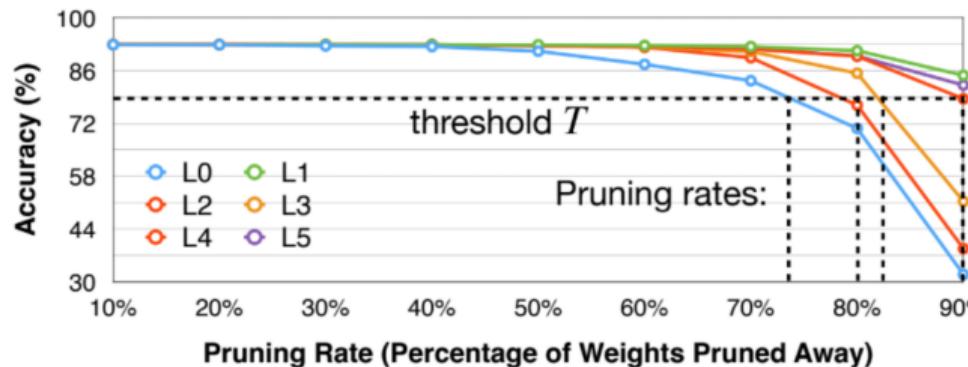
- The process of Sensitivity Analysis (*VGG-11 on CIFAR-10 dataset)
 - Pick a layer L_i in the model
 - Prune the layer L_i with pruning ratio $r \in \{0, 0.1, 0.2, \dots, 0.9\}$ (or other strides)
 - Observe the accuracy degrade ΔAcc_r^i for each pruning ratio
 - Repeat the process for all layers



Finding Pruning Ratios

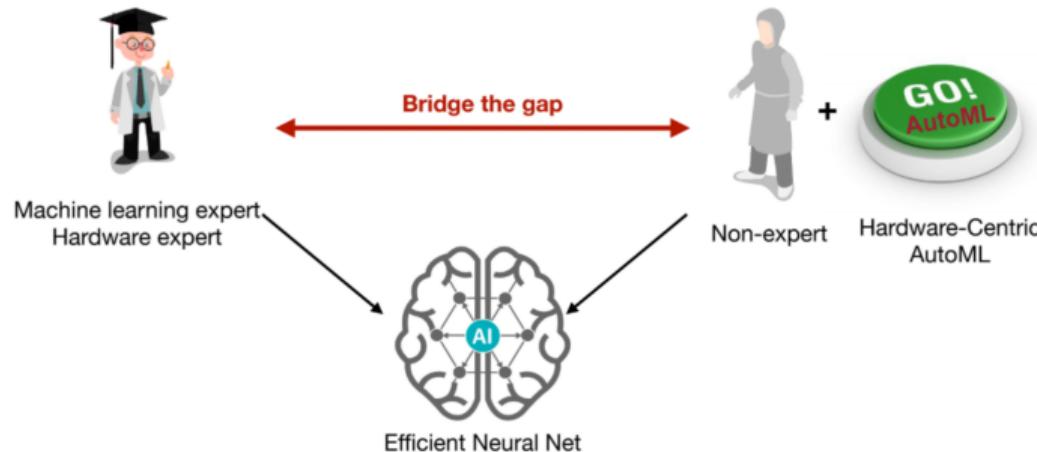
Analyse the sensitivity of each layer

- The process of Sensitivity Analysis (*VGG-11 on CIFAR-10 dataset)
 - Pick a layer L_i in the model
 - Prune the layer L_i with pruning ratio $r \in \{0, 0.1, 0.2, \dots, 0.9\}$ (or other strides)
 - Observe the accuracy degrade ΔAcc_r^i for each pruning ratio
 - Repeat the process for all layers
 - Pick a degradation threshold T such that the overall pruning rate is desired



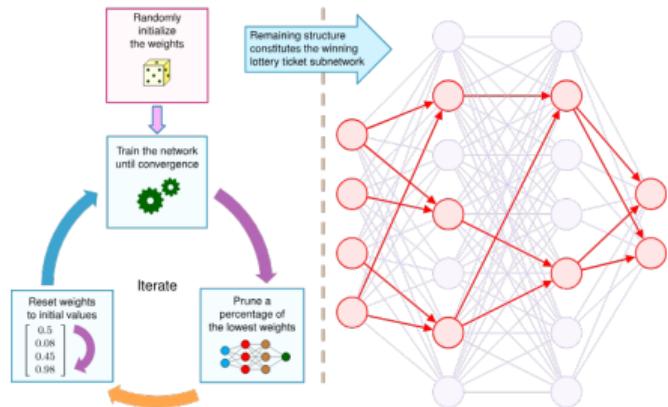
Automatic Pruning

Can we have a push-the-button solution?



Lottery Ticket Hypothesis

- **Core Idea:** Large neural networks contain small, sparse subnetworks called "winning tickets."
- These "winning tickets" can be trained in isolation from the start to achieve the same (or better) accuracy as the full, dense network.
- LTH reframes pruning not just as compression, but as a discovery mechanism to find these inherently efficient subnetworks.⁵

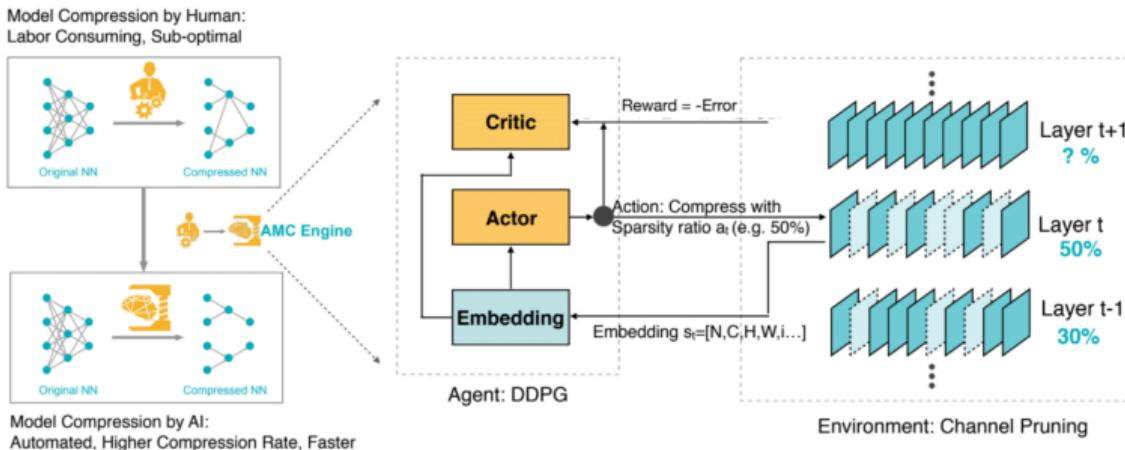


- **LTH challenges overparameterization:** It suggests that massive models may not be necessary for training, but rather to ensure a "winning ticket" exists at initialization.
- **Practical Example:** A 90% pruned BERT-base (a "winning ticket") can retain 97% of its performance and train 5-8x faster.

⁵Frankle, Jonathan, and Michael Carbin. "The lottery ticket hypothesis: Finding sparse, trainable neural networks." arXiv preprint arXiv:1803.03635 (2018).

AMC: AutoML for Model Compression

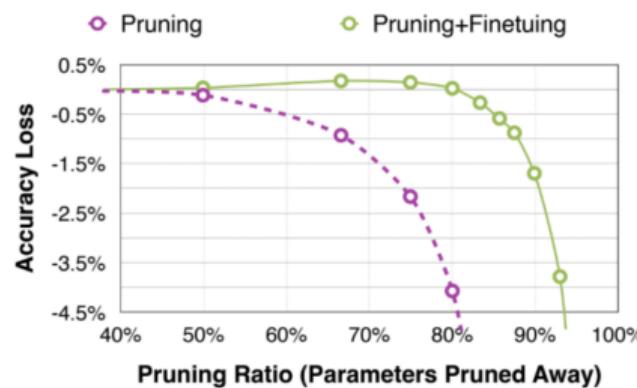
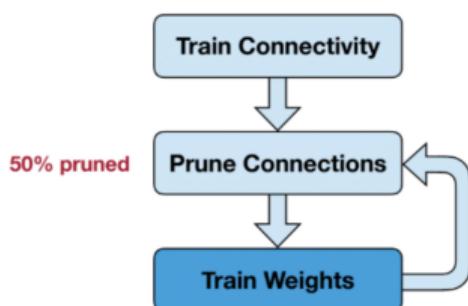
Pruning as a reinforcement learning problem⁶



⁶He, Yihui, et al. "AMC: Automl for model compression and acceleration on mobile devices." Proceedings of the European conference on computer vision (ECCV). 2018.

Fine-tuning Pruned Neural Networks

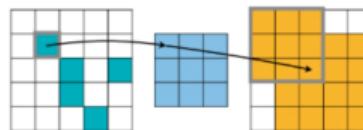
- After pruning, the model may decrease, especially for larger pruning ratio.
- Fine-tuning the pruned neural networks will help recover the accuracy and push the pruning ratio higher.
- Consider pruning followed by a fine-tuning is one iteration.
- **Iterative Pruning:** gradually increases the target sparsity in each iteration.



Sparsity

Sparse convolution computation

Conventional Convolution



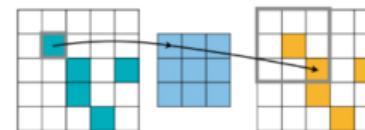
Maps
 (In, Out, Wgt)

$(P_0, Q_0, W_{1,1})$
 $(P_0, Q_1, W_{1,0})$
 $(P_0, Q_2, W_{1,-1})$
 $(P_0, Q_3, W_{0,1})$
 $(P_0, Q_4, W_{0,0})$
 $(P_0, Q_5, W_{0,-1})$
 $(P_0, Q_6, W_{-1,1})$
 $(P_0, Q_7, W_{-1,0})$
 $(P_0, Q_8, W_{-1,-1})$

Computation
 $(f_{out} = f_{out} + f_{in} \times W_{Wgt})$ for
each entry in the maps

9 matrix multiplications

Sparse Convolution



No compute
No compute
No compute
No compute
 $(P_0, Q_0, W_{0,0})$
No compute
No compute
No compute
 $(P_0, Q_1, W_{-1,-1})$

2 matrix multiplications

Hands-on



“Talk is cheap. Show me the code.”

— Linus Torvalds