Neural Networks Speed-up and Compression

Lecture 2: Pruning

Recap from Lecture 1

We have considered

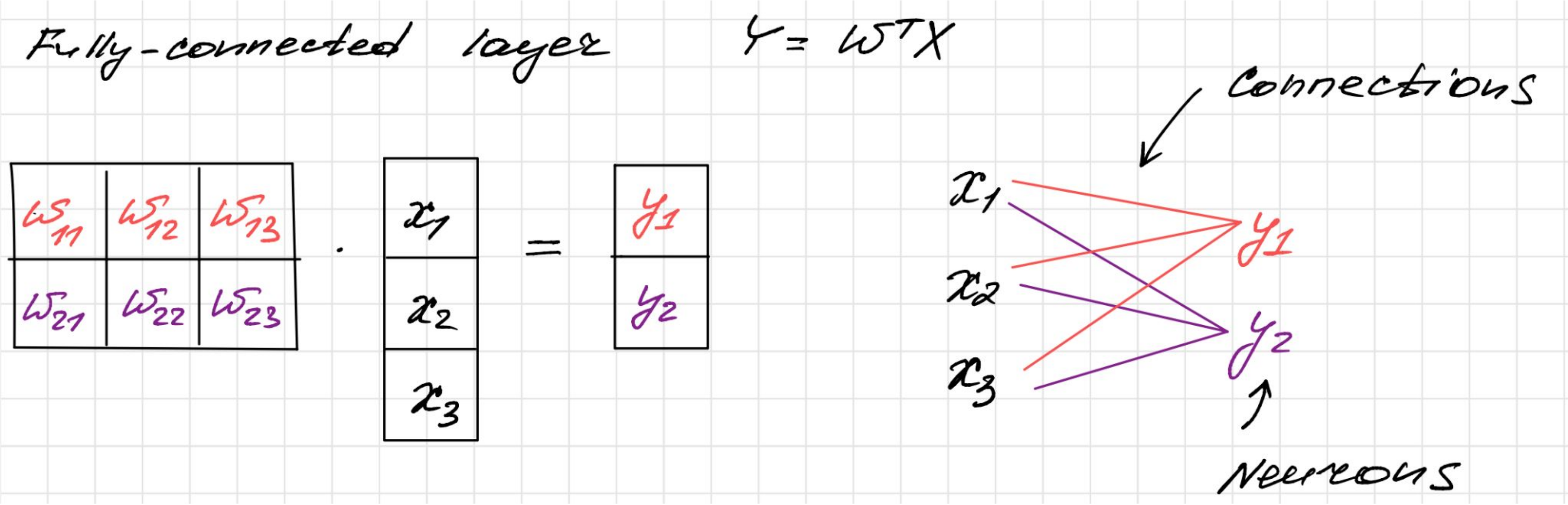
- Main neural network layers (linear, non-linear, normalization)
- Theoretical measurements of speed and memory
  - FLOP / MAC
  - number of parameters
- Empirical measurements of speed and memory
  - FlopCo
  - PyTorch tools
- Arithmetic intensity

Outline

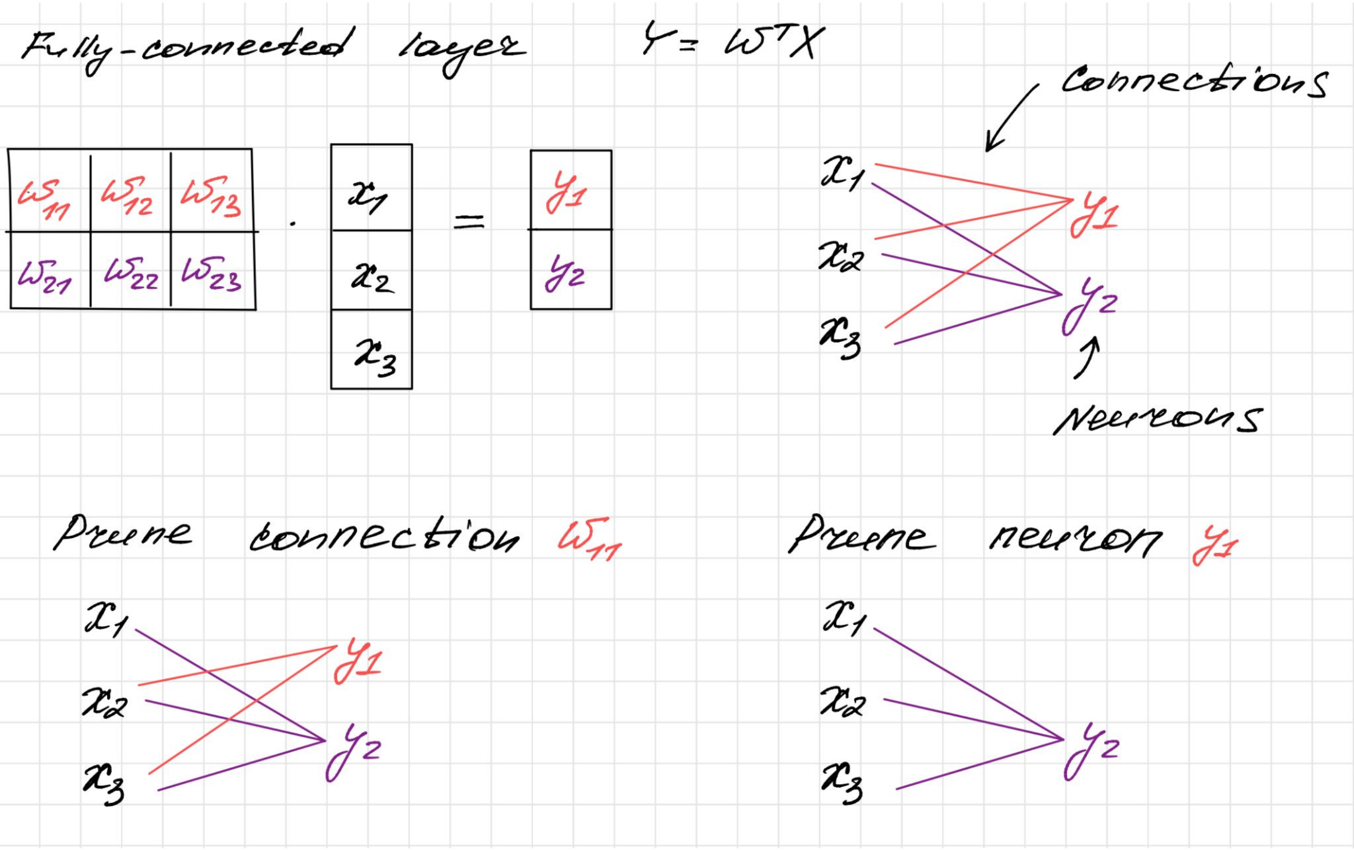
- Neurons and connections in neural networks
    - Fully-connected & Convolutional layers
    - Pruning
- Unstructured pruning
- Structured pruning

Neurons and Connections

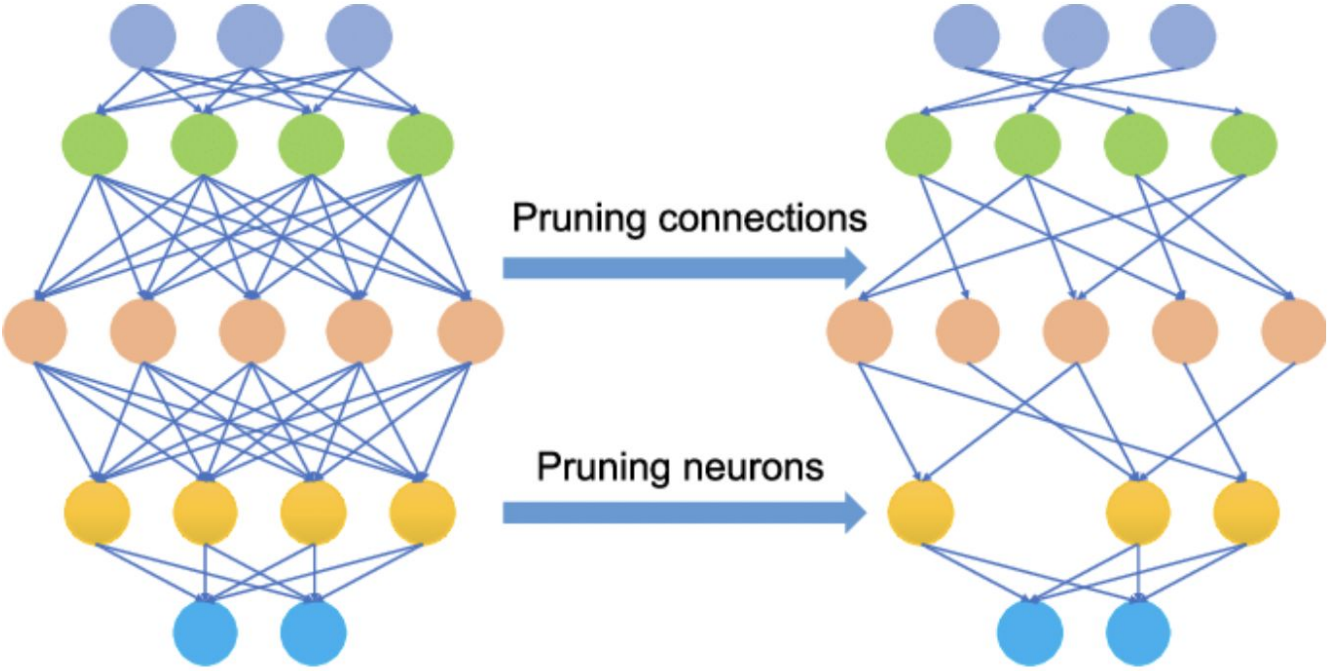
Fully-Connected Layer



Fully-Connected Layer: Prune



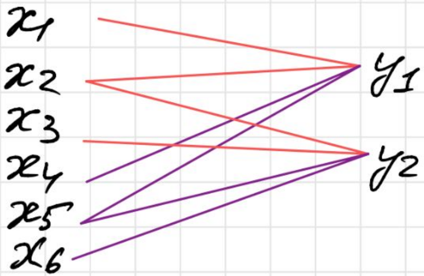
Fully-Connected Neural Network



Convolutional Layer

Convolutional layer $\qquad Y = X * W$

$$\begin{bmatrix} x_1 & x_2 & x_3 \\ x_4 & x_5 & x_6 \\ x_7 & x_8 & x_9 \end{bmatrix} * \begin{bmatrix} w_{11} & w_{12} \\ w_{21} & w_{22} \end{bmatrix} = \begin{bmatrix} y_1 = w_{11}x_1 + w_{12}x_2 + w_{21}x_4 + w_{22}x_5 & y_2 = w_{11}x_2 + w_{12}x_3 + w_{21}x_5 + w_{22}x_6 \\ y_3 = w_{11}x_4 + w_{12}x_5 + w_{21}x_7 + w_{22}x_8 & y_4 = w_{11}x_5 + w_{12}x_6 + w_{21}x_8 + w_{22}x_9 \end{bmatrix}$$

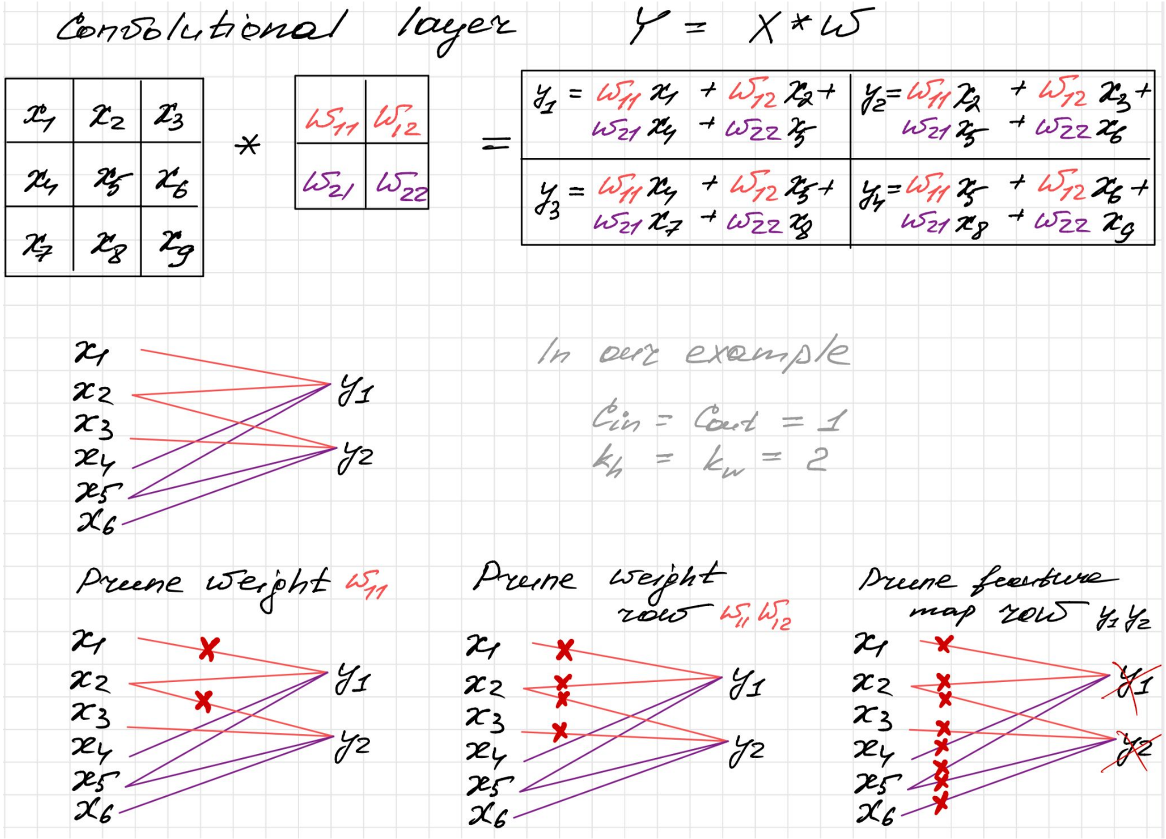


In our example

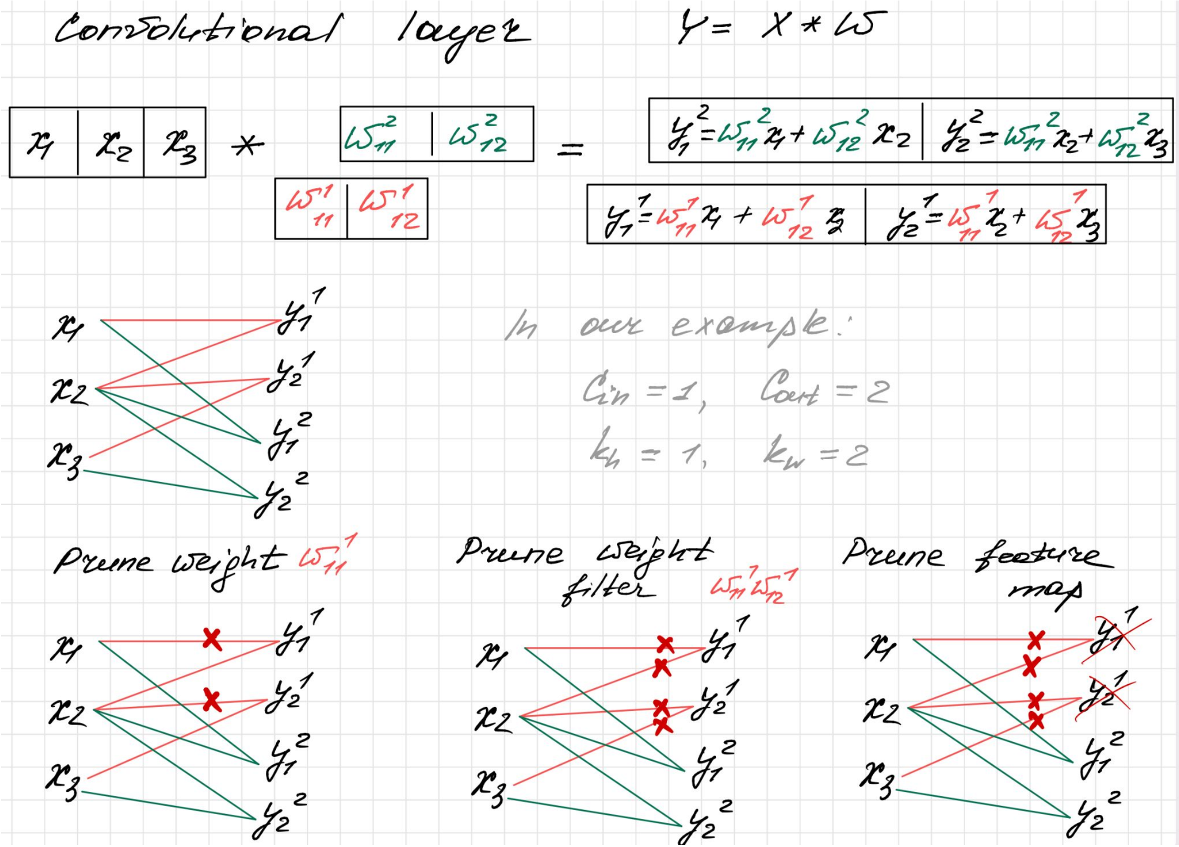
$C_{in} = C_{out} = 1$

$k_h = k_w = 2$

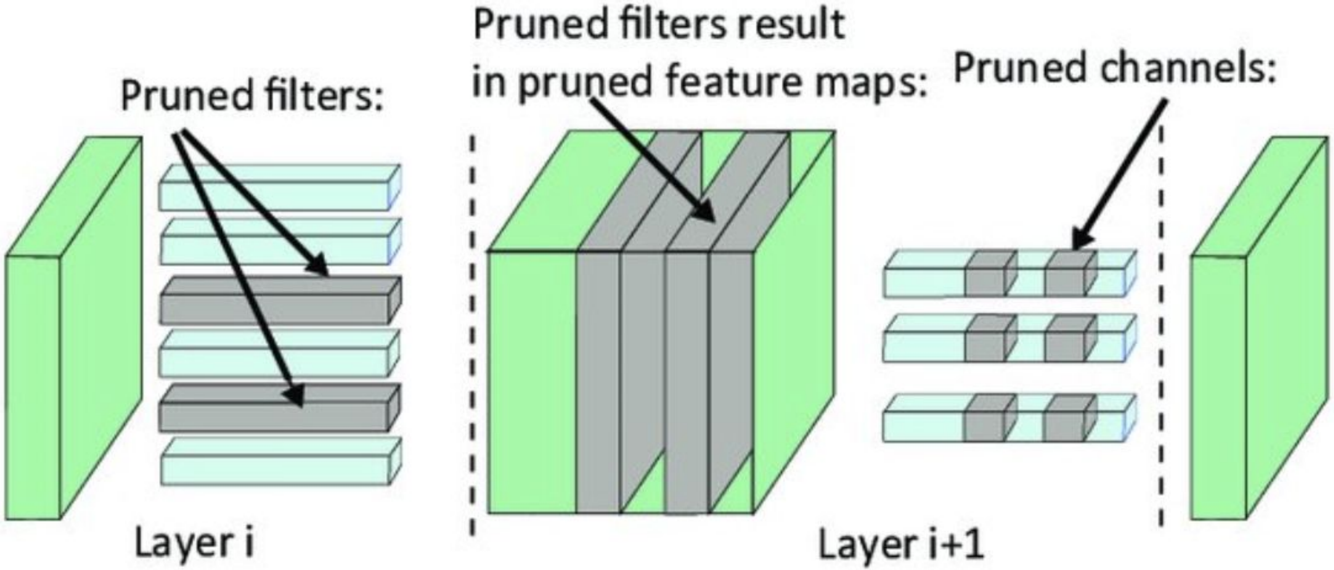
Convolutional Layer: Prune Shape (Row)



Convolutional Layer: Prune Filter

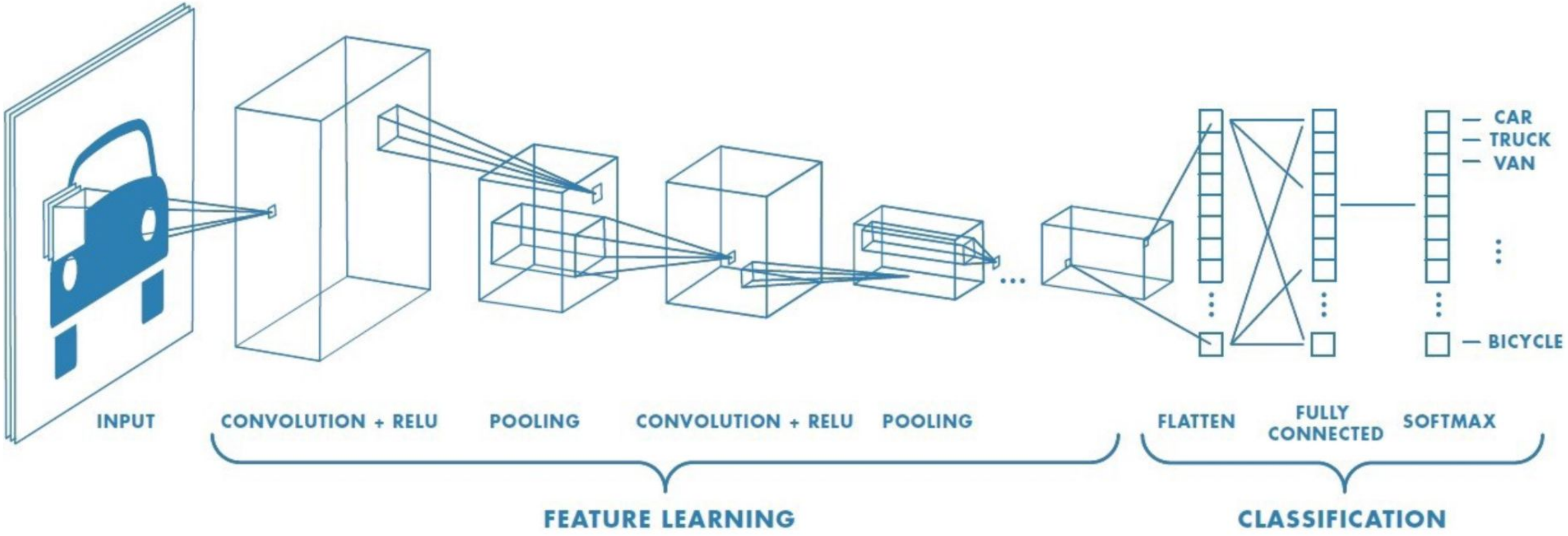


Convolutional Neural Network

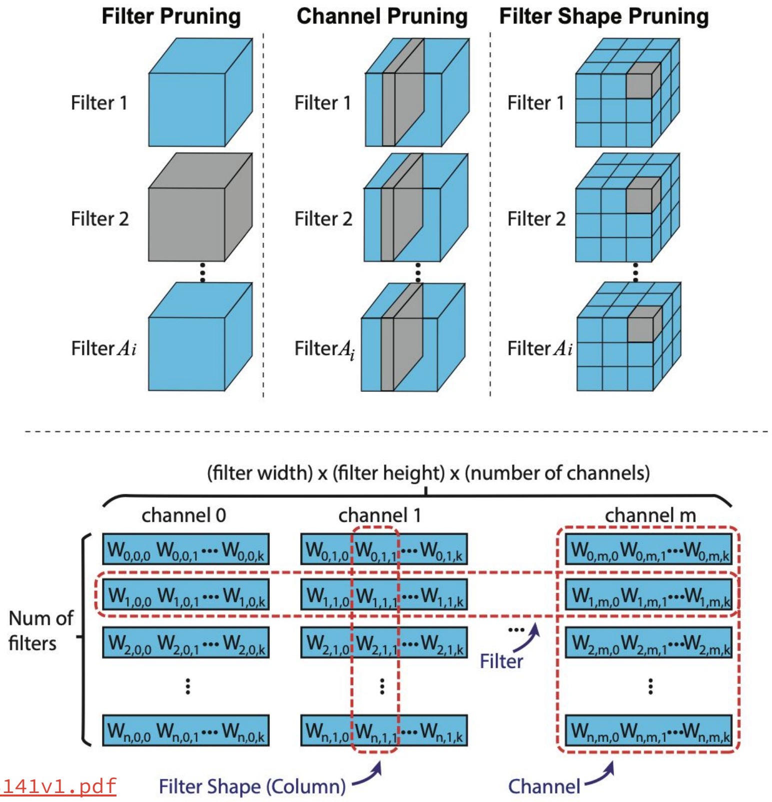


Source: https://www.researchgate.net/publication/334248569_Non-structured_DNN_Weight_Pruning_Considered_Harmful

Convolutional Neural Network



Convolutional Layer: Prune Filter/Channel/Shape



Source: https://arxiv.org/pdf/1907.03141v1.pdf

What's the problem?

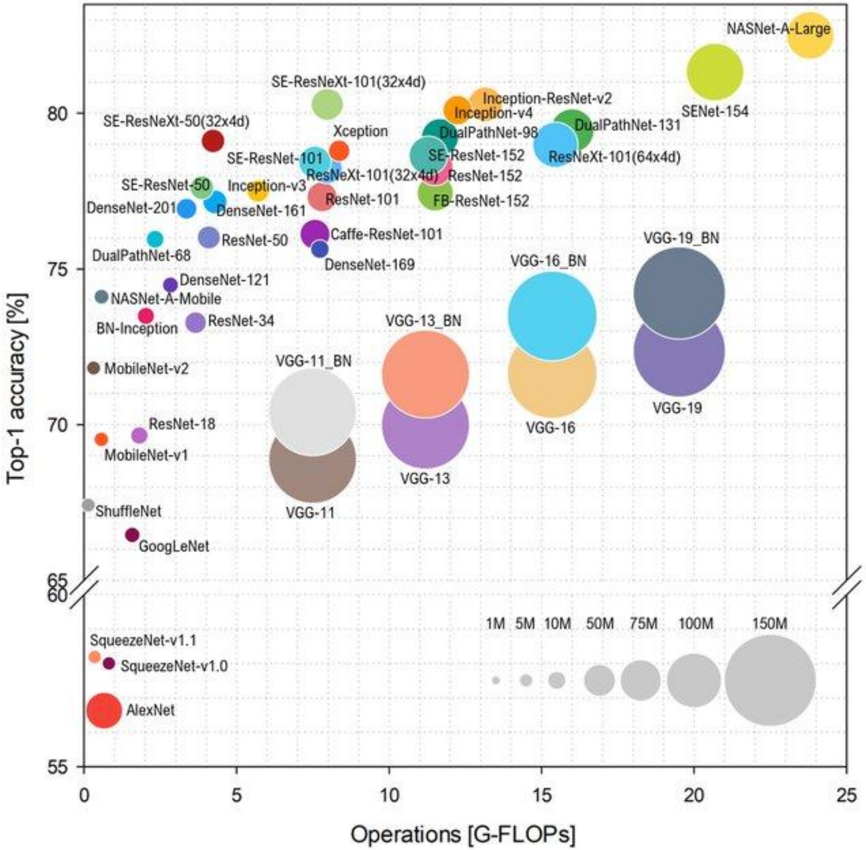
DL model limitations:

- High memory consumption
- Huge computational requirements
- Great power consumption

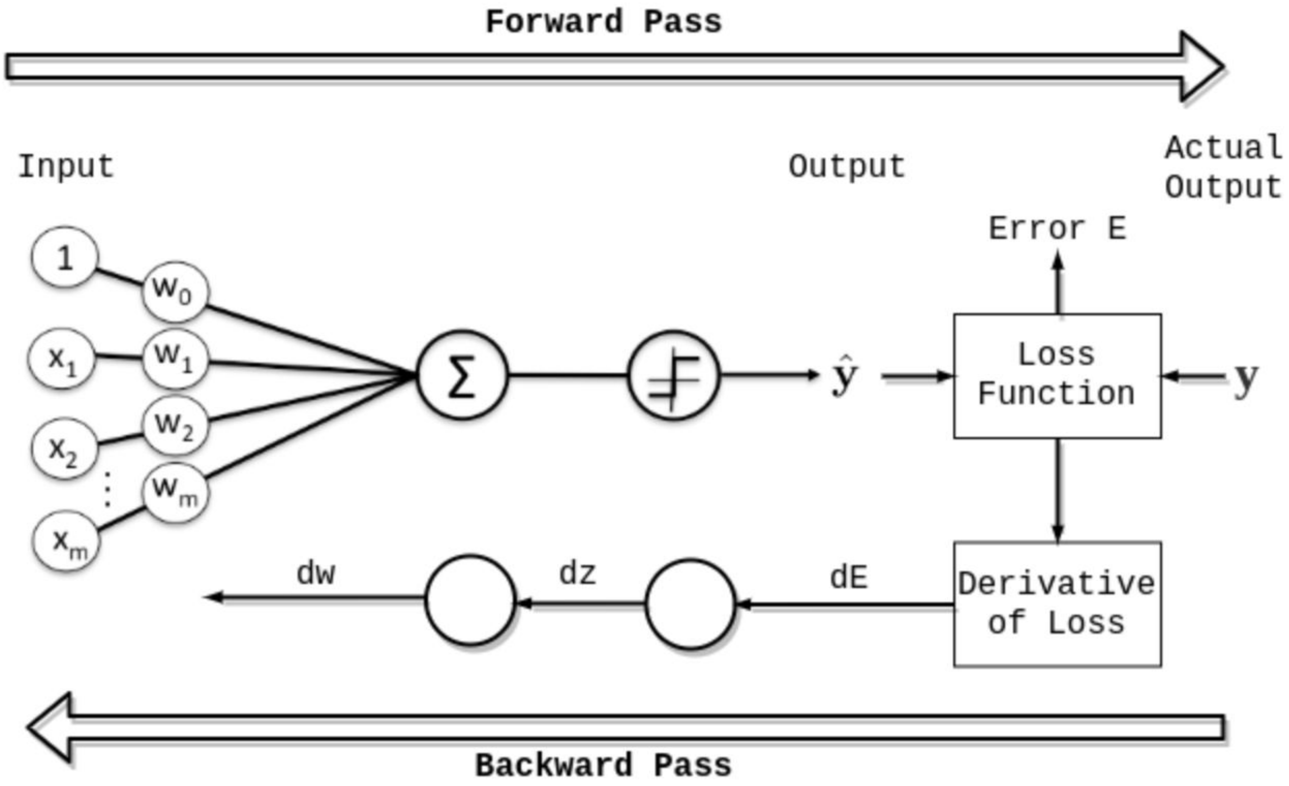
Difficult to deploy on portable devices

(e.g. laptops and smartphones)

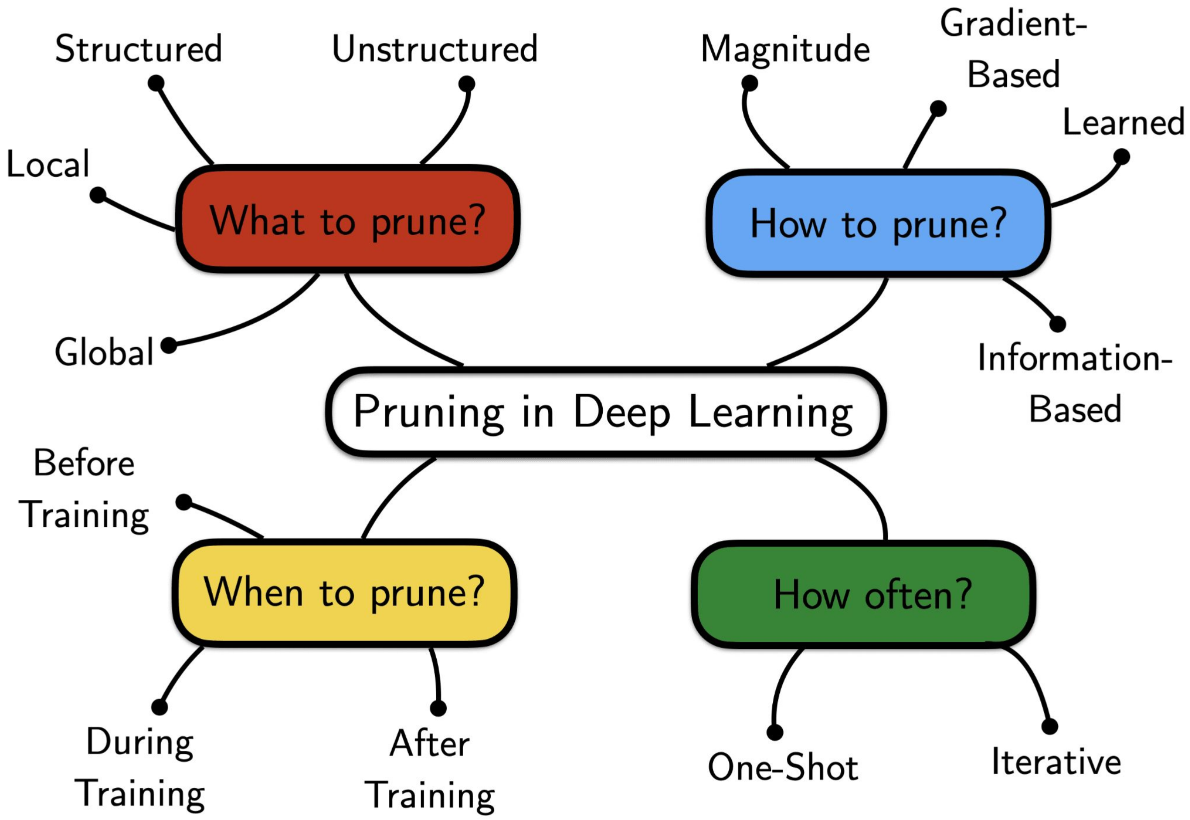
Efficient architecture design is required



Recap: Epoch of Neural Network Training



Pruning

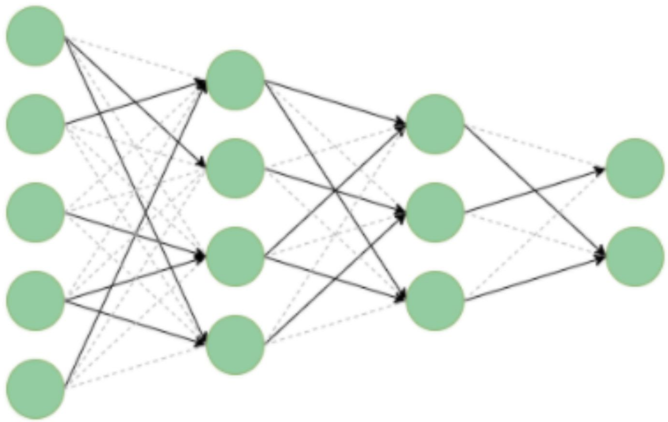


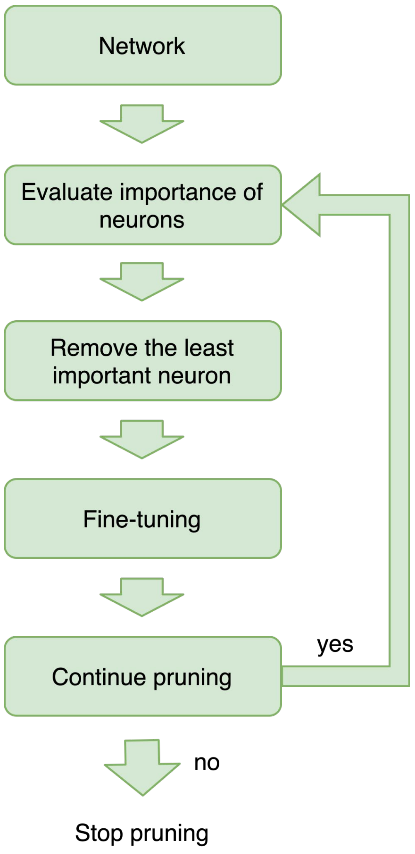
Structured
Unstructured
Magnitude
Gradient-Based
Learned
Local
What to prune?
How to prune?
Global
Information-Based
Pruning in Deep Learning
Before Training
When to prune?
How often?
During Training
After Training
One-Shot
Iterative

Unstructured Pruning

Unstructured Pruning

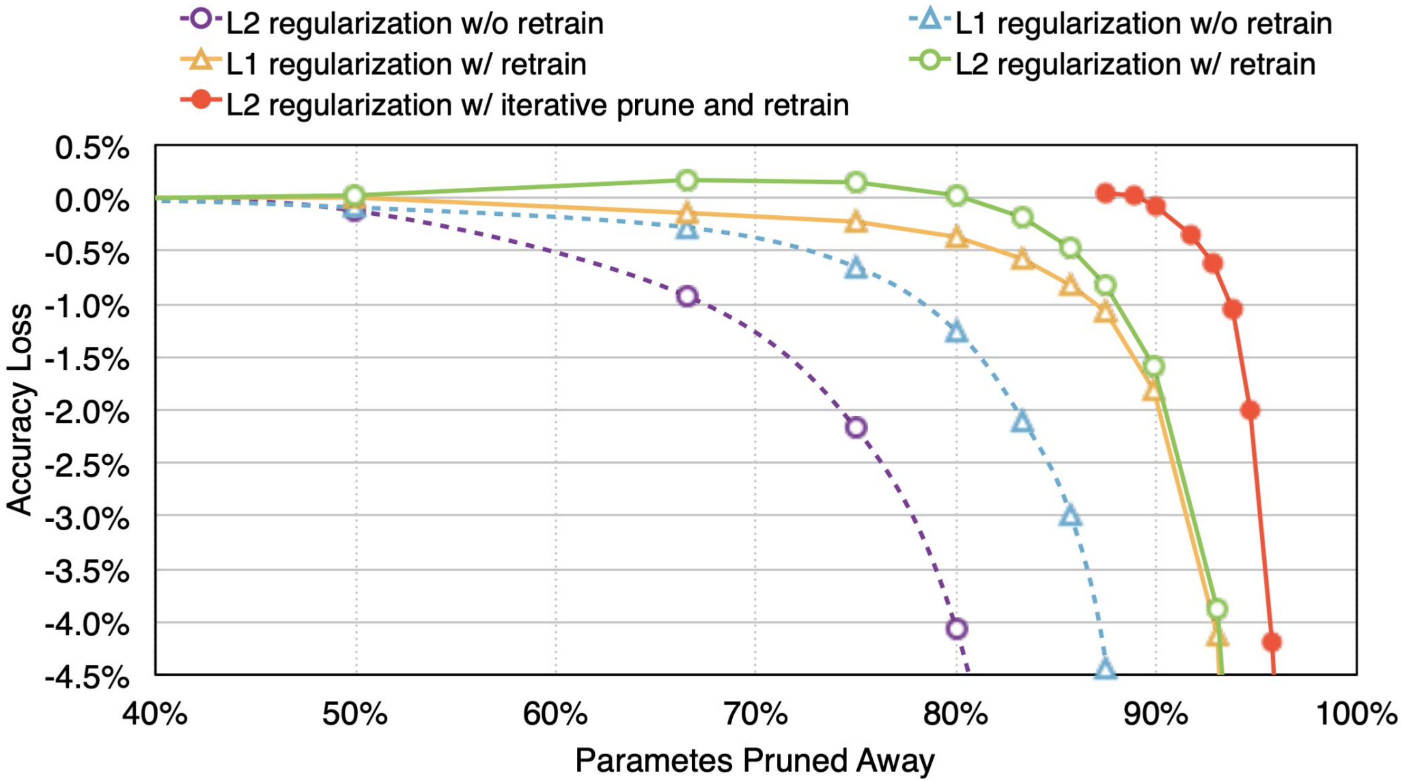
or Fine-Grained Pruning or Weight Sparsification





Source: https://arxiv.org/pdf/1506.02626.pdf

Unstructured Pruning



Source: https://arxiv.org/pdf/1506.02626.pdf

Unstructured Pruning

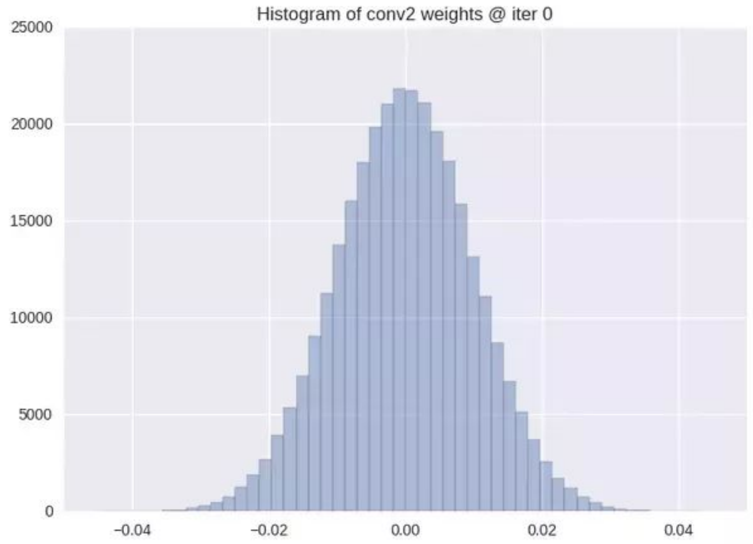
List of possible criterions:

- Weight-based criteria (L1/L2 norm)

- Gradient-based criteria

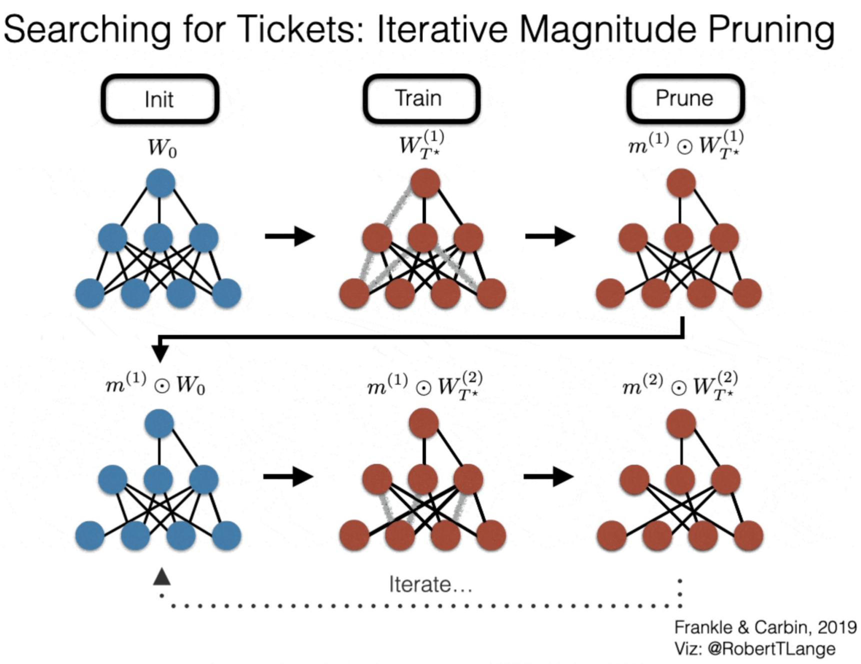
$$\delta E = \sum_i g_i \delta u_i + \frac{1}{2}\sum_i h_{ii}\delta u_i^2 + \frac{1}{2}\sum_{i \neq j} h_{ij}\delta u_i \delta u_j + O(||\delta U||^3)$$

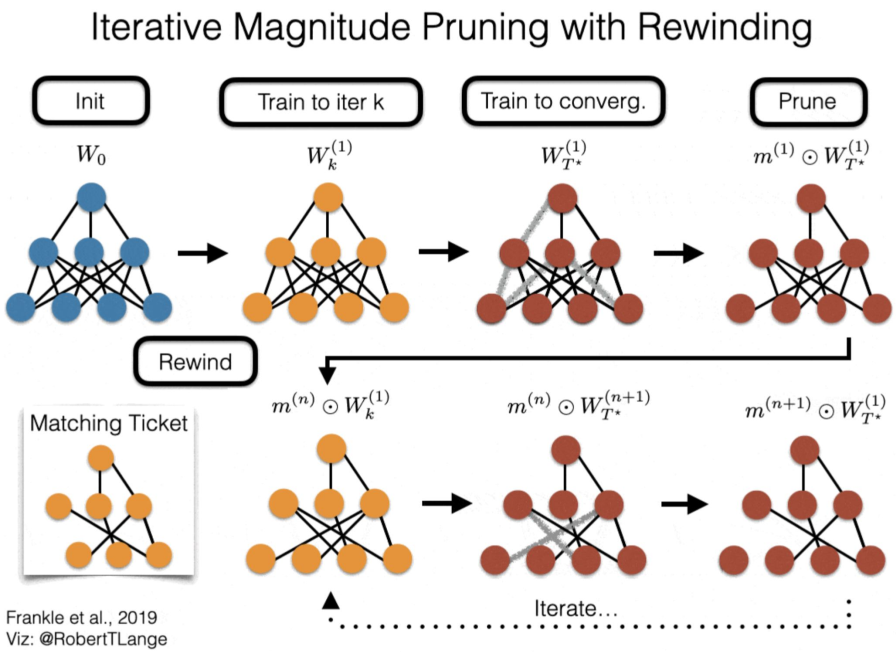
$$g_i = \frac{\partial E}{\partial u_i} \quad \text{and} \quad h_{ij} = \frac{\partial^2 E}{\partial u_i \partial u_j}$$



Source: https://openaccess.thecvf.com/content_CVPR_2020/papers/He_Learning_Filter_Pruning_Criteria_for_Deep_Convolutional_Neural_Networks_Acceleration_CVPR_2020_paper.pdf

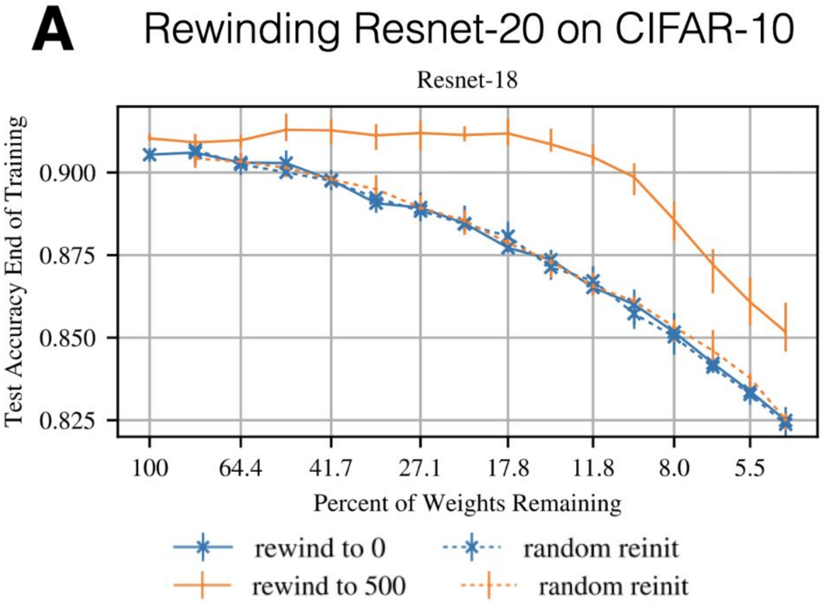
Unstructured Pruning: Lottery Ticket Hypothesis

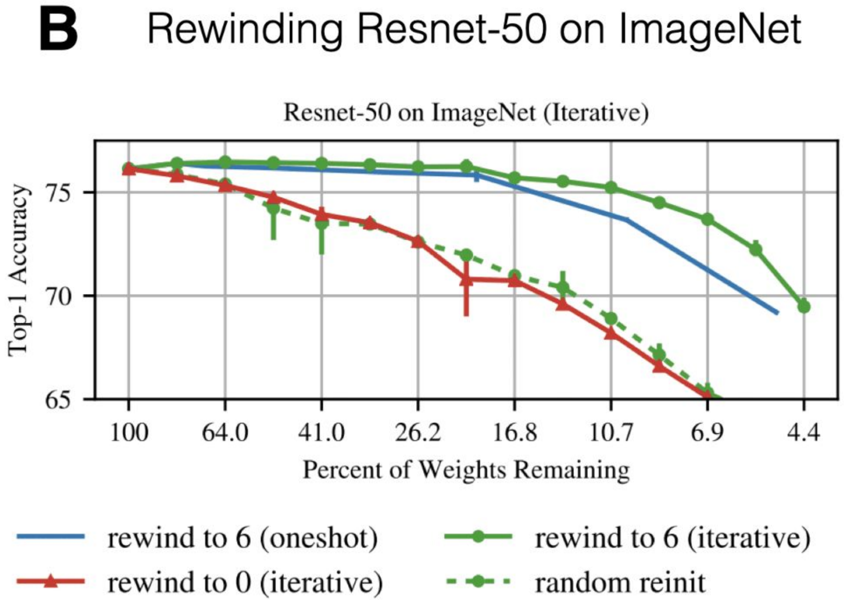




Source: https://towardsdatascience.com/the-lottery-ticket-hypothesis-a-survey-d1f0f62f8884
https://arxiv.org/pdf/1803.03635.pdf
https://arxiv.org/pdf/1903.01611.pdf

Unstructured Pruning: Lottery Ticket Hypothesis





Source: https://towardsdatascience.com/the-lottery-ticket-hypothesis-a-survey-d1f0f62f8884

Unstructured Pruning

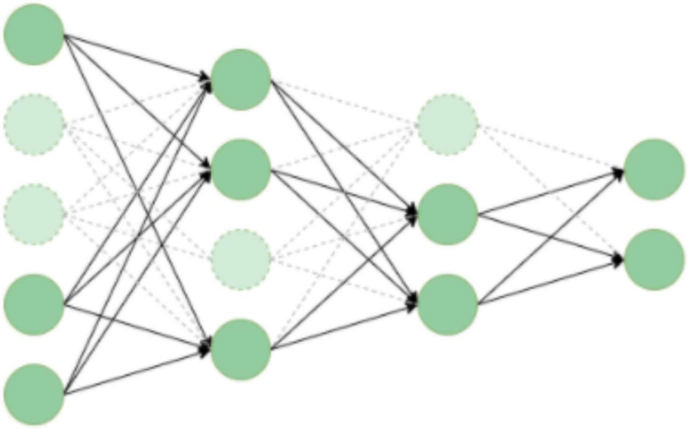
- Benefits
    - Achieves high rates of weight reduction without acc. drop (~ 95% of weights can be removed)

- Drawbacks
    - Requires hardware support for sparse computation speedup
    - Hard to find sparsity level for all layers of the network

Structured Pruning

Structured Pruning

Removing structural parts instead of individual weights



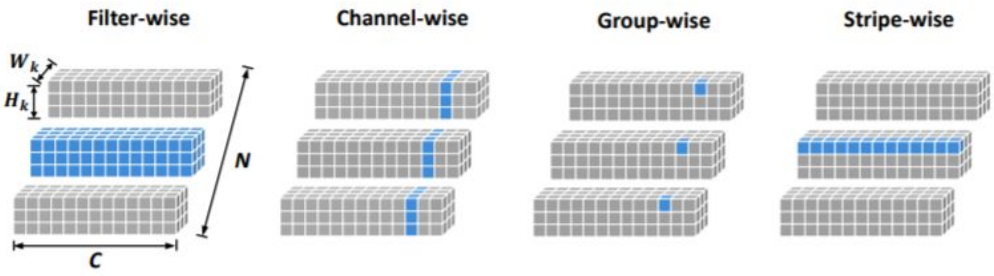


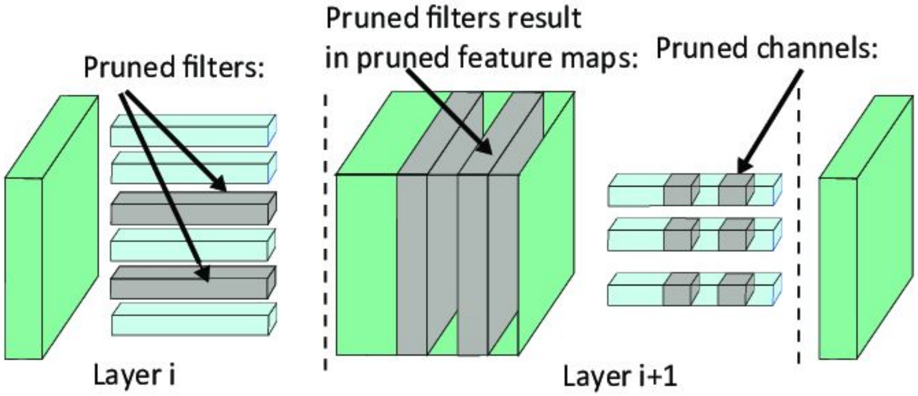
Figure 2: The visualization of different types of pruning.

Source: https://heartbeat.fritz.ai/neural-network-pruning-research-review-2020-bc21a77f0295

Structured Pruning

List of possible criterions:

- Weight-based criteria:
  - L1/L2 norm
  - Scaling parameter in BN
- Activation-based criteria:
  - PCA of activations
- Gradient-based criteria
- Greedy and One-shot Pruning



Source: https://openaccess.thecvf.com/content_CVPR_2020/papers/He_Learning_Filter_Pruning_Criteria_for_Deep_Convolutional_Neural_Networks_Acceleration_CVPR_2020_paper.pdf

Structured Pruning

- Benefits
  - Efficiently accelerates model
  - No need of hardware/software support (pruned model is structurally equivalent to initial model)

- Drawbacks
  - Pruning channels/filters in one layer affects previous/subsequent layers
  - Hard to find best filter configuration for the whole network

Resume

Resume

Current pipeline in pruning can be divided into two components

- Identifying the most promising neurons to be pruned
- Training and fine-tuning the pruned model to recover the base model's prediction performance

A successful pruning algorithm is an iterative progression of these components

Data: CNN model, training data
while *Compression requirement not met* **or** *exceed budget*
do
  train model (to convergence);
  compute pruning criteria;
  prune parameters below threshold;
end

Algorithm 1: Workflow for model pruning.

Resume

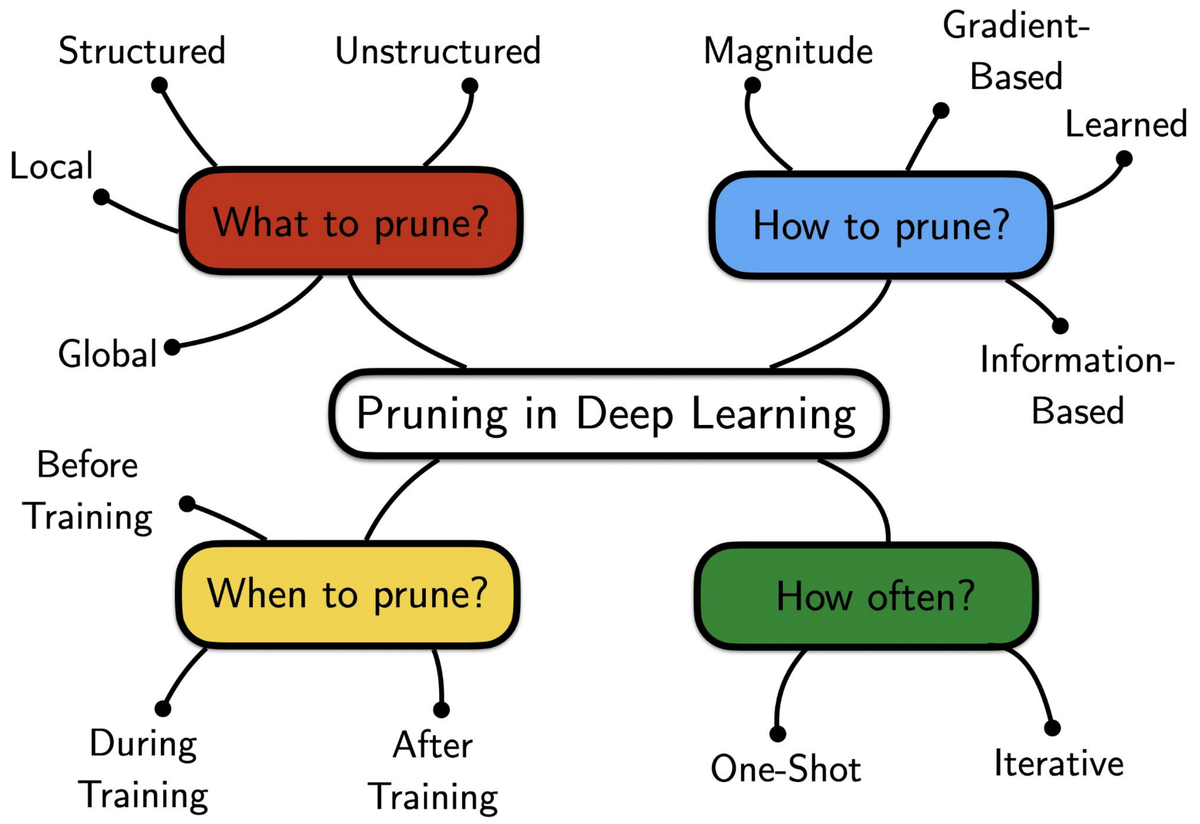
- Data-agnostic
- Data-aware

Table 1: Saliency measurements used in pruning.

Basic idea	Saliency Expression
Minimize pruned deterioration	$H_{ii}w_i^2$; w_i^2/H_{ii}^{-1}
Remove weights with small values	$\|w\|$
Weight similarity and redundancy	$\|\|w_i - w_j\|\|$; $\|\|w - \mathbb{E}_{\text{geo}}[w]\|\|$
Structured L1/2-norm penalty	Penalize $\sum_{c'} w^2_{c',c,i,j}$; $\sum_{c,i,j} w^2_{c',c,i,j}$; $\sum_{c'} w^2_{c',c,i,j}$; $\sum_{c,i,j} \|w_{c',c,i,j}\|$
Magnitude of Batch Norm	-
Remove by filter similarity	Geometric mean
Remove inactive neurons (APoZ)	$\sum_i \mathbb{I}_{z_i=0}$
Remove activations with flat gradient	$\|\sum_i \frac{\mathcal{L}}{\partial z_i} z_i\|$
Reconstruction error on channel pruning	$\|\|z_i - z_i^{\text{pruned}}\|\|$
Entropy	$\sum_m P(z_i)\log P(z_i)$, where $P(z_i)$ is probability of activations in bins
Reconstruction error and L1 norm	-
Neural Importance Score	$s_k = \|w_{i+1}\|^T s_{k+1}$
Remove insensitive neurons	Reset weights with small updates to initial value

Source: https://arxiv.org/pdf/2005.04275.pdf

Pruning


Structured
Unstructured
Magnitude
Gradient-Based
Learned
Local
What to prune?
How to prune?
Global
Information-Based
Pruning in Deep Learning
Before Training
When to prune?
How often?
During Training
After Training
One-Shot
Iterative