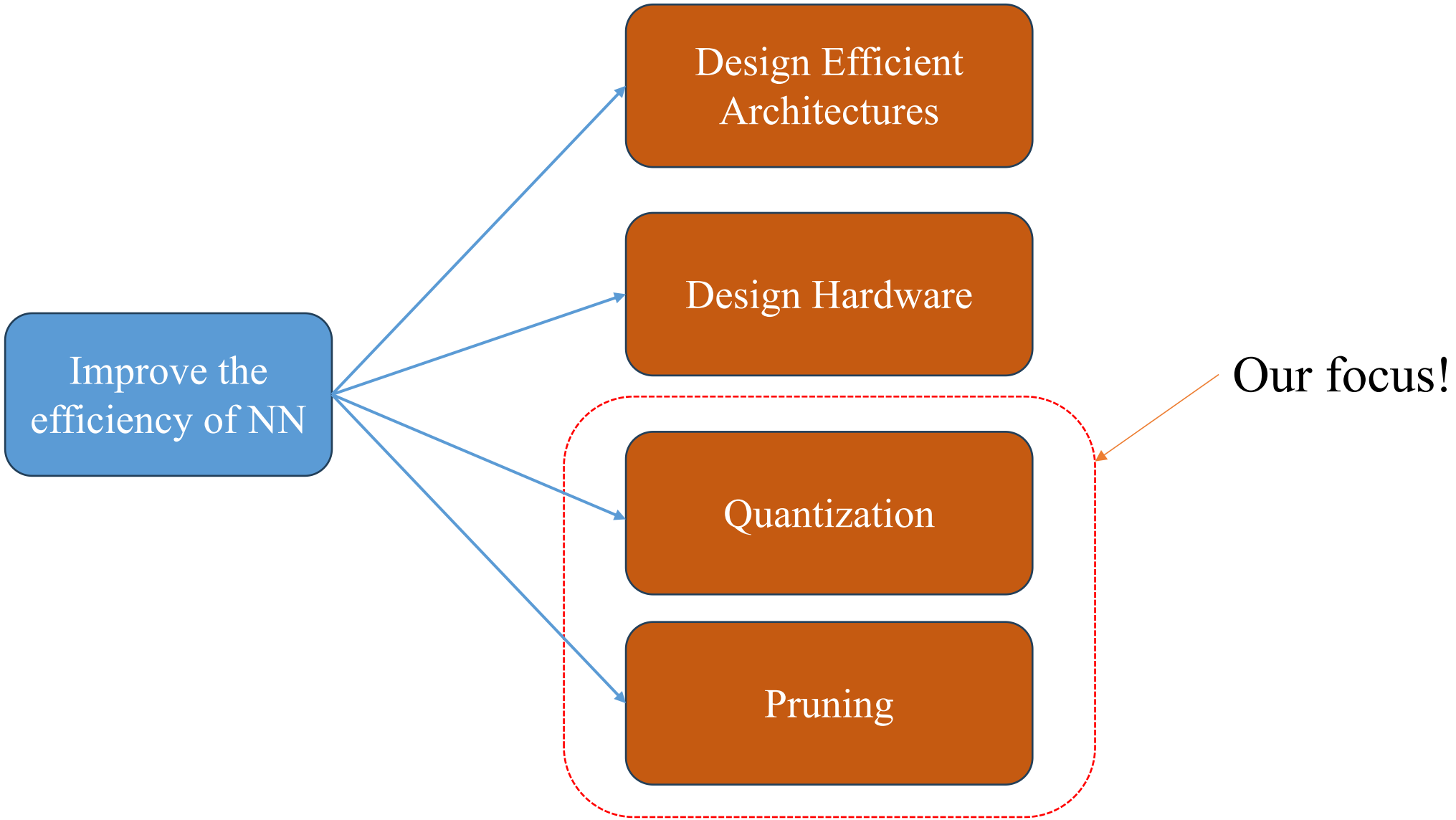
AI VIETNAM
All-in-One Course

Quantization + Pruning

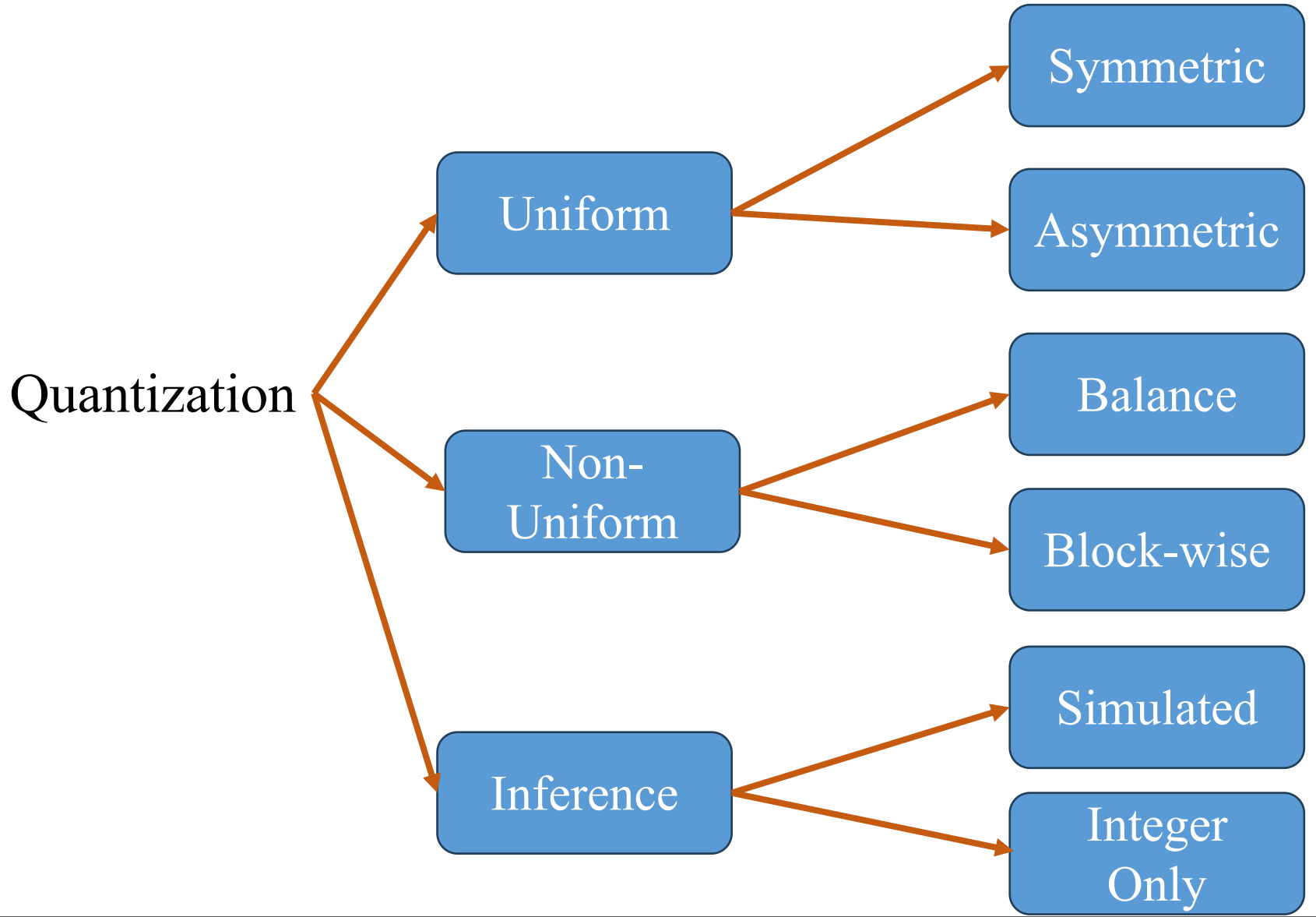
Bach-Hoang Ngo

Year

Motivation


Improve the efficiency of NN
Design Efficient Architectures
Design Hardware
Quantization
Pruning
Our focus!

Summary

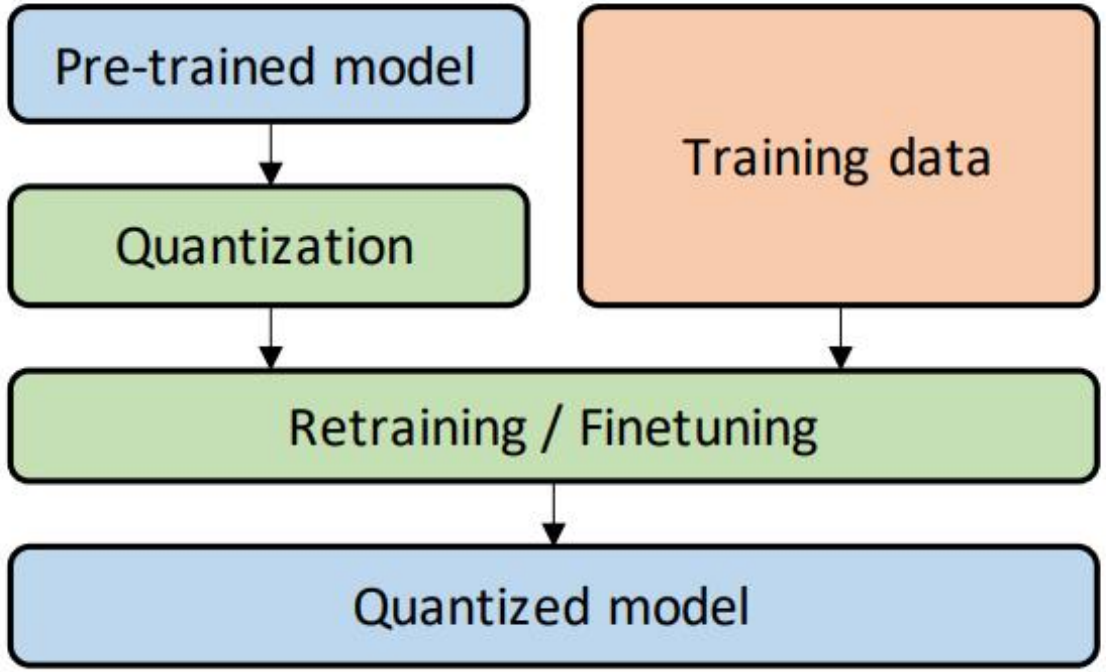

Quantization
Uniform
Symmetric
Asymmetric
Non-
Uniform
Balance
Block-wise
Inference
Simulated
Integer
Only

AI VIETNAM
All-in-One Course

Quantization

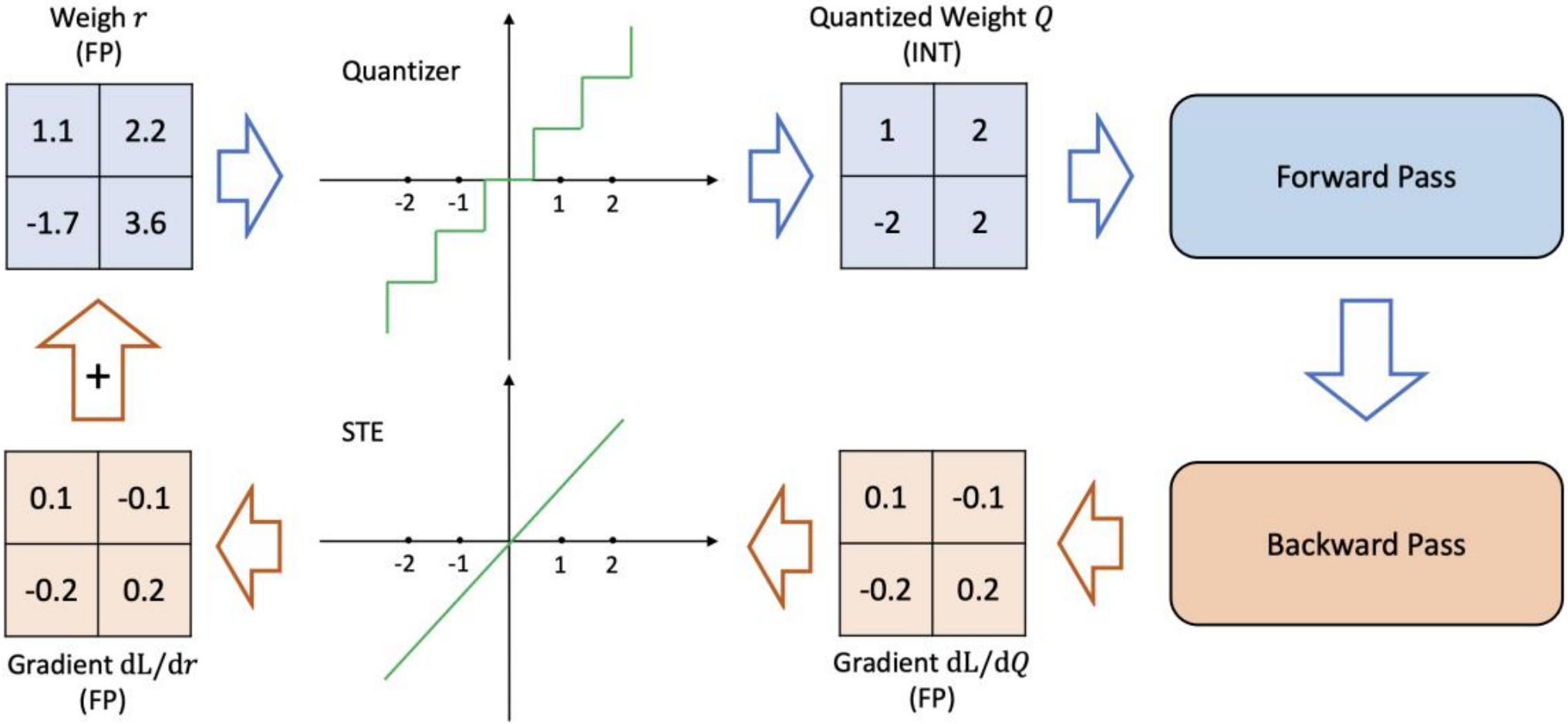
- Floating Point
- Quantization
- **Quantization Aware Training**
- Post Training Quantization

Quantization Aware Training



Naïve Quantization -> degradation in model accuracy
QAT -> Simulating the quantization effects during training

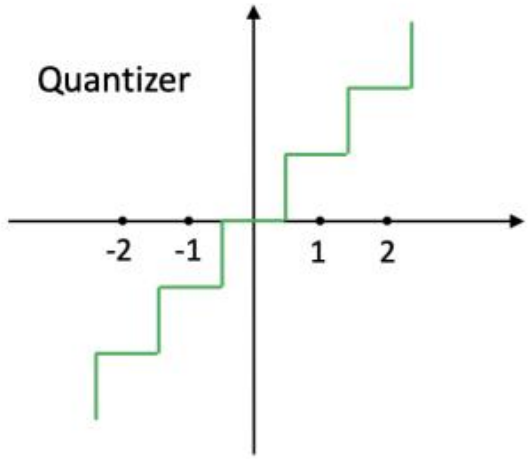
Quantization Aware Training



Quantization Aware Training

Weight (FP)

1.1	2.2
-1.7	3.6



Quantized Weight

1	2
-2	2

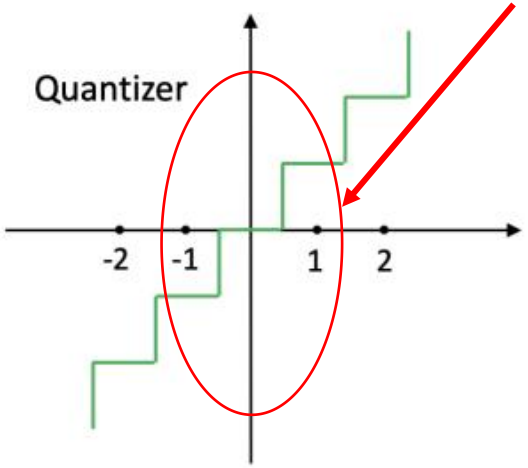
Forward pass

Quantization Aware Training

Non differentiable

Weight (FP)

1.1	2.2
-1.7	3.6



Quantized Weight

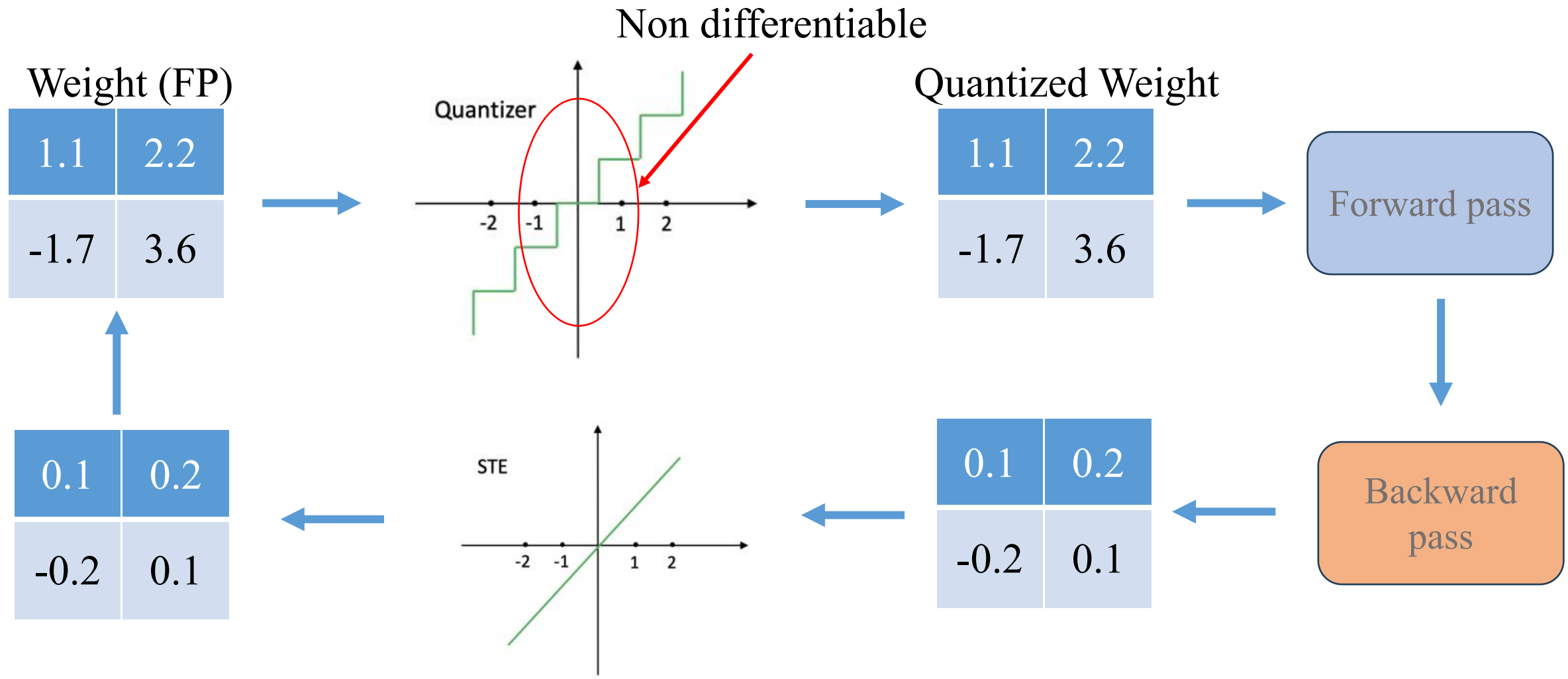
1	2
-2	2

Forward pass

Backward pass

0.1	0.2
-0.2	0.1

Quantization Aware Training

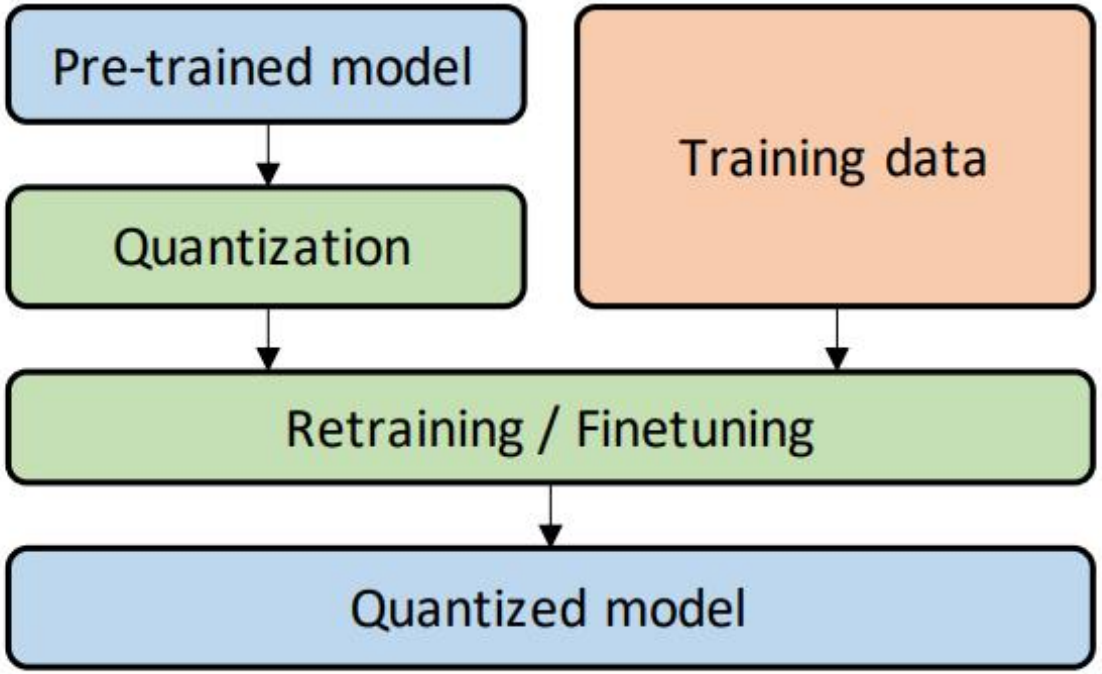


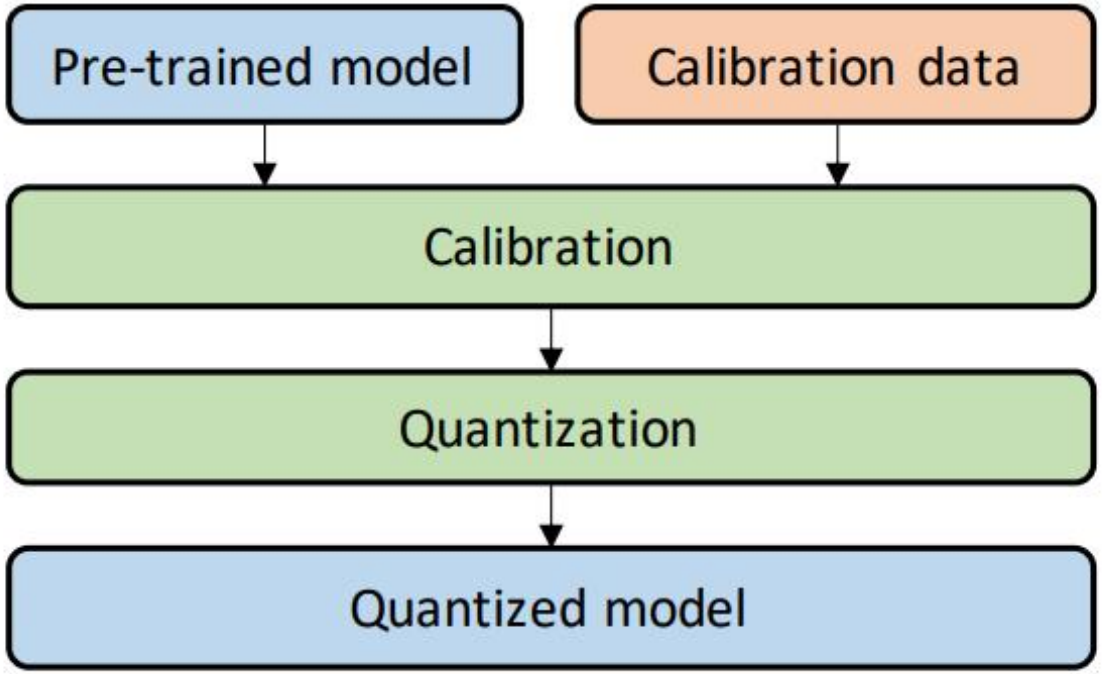
AI VIETNAM
All-in-One Course

Quantization

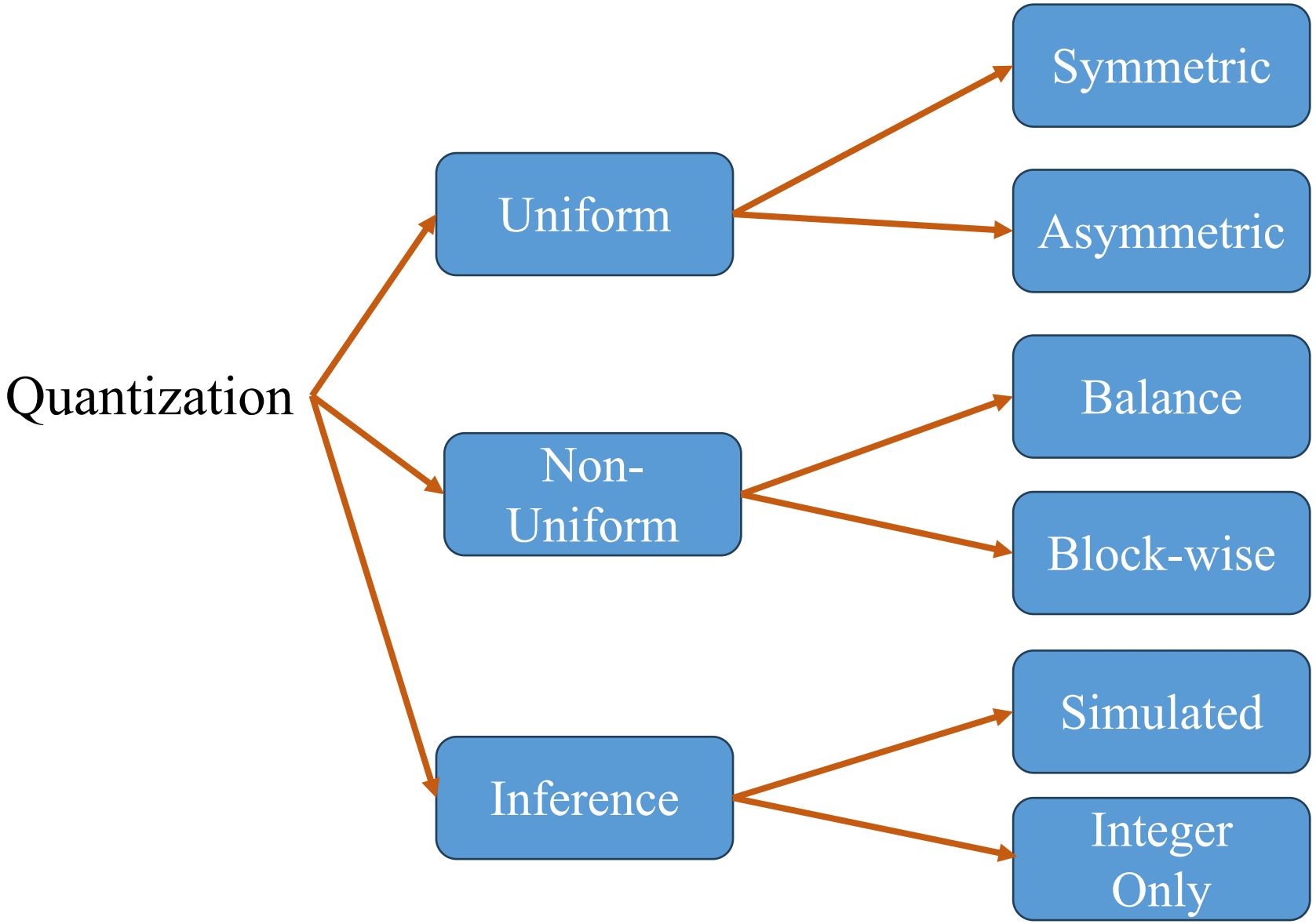
- Floating Point
- Quantization
- Quantization Aware Training
- Post Training Quantization

Post-Training Quantization





Post Training Quantization



Post-Training Quantization

Feature	PTQ (Post-Training Quantization)	QAT (Quantization-Aware Training)
Model Size Reduction	Effective in reducing model size	Can achieve similar or slightly better size reduction compared to PTQ
Inference Speed	Improves inference speed due to lower precision calculations	Can lead to even faster inference speed compared to PTQ
Accuracy	May experience larger accuracy degradation	Generally preserves accuracy better than PTQ
Training Complexity	Simpler to implement, requires minimal modification	More complex to implement, requires modifying training loop

Quantization Memory

7B LLM -> OOM

```
[ ]  1 # using huggingface from_pretrained
     2 model_name = 'vilm/vinallama-7b-chat'
     3 model = AutoModel.from_pretrained(model_name, device_map='cuda') # lead to OOM in cuda mem
```

4bit quantization -> OK

```
[ ]  1 # using bitsandbyte
     2 model_name = 'vilm/vinallama-7b-chat'
     3 device_map = {
     4     "transformer.word_embeddings": 0, # 0 mean gpu
     5     "transformer.word_embeddings_layernorm": 0,
     6     "lm_head": "cpu", # offload lm_head to cpu
     7     "transformer.h": 0,
     8     "transformer.ln_f": 0,
     9 }
    10 quantization_config = BitsAndBytesConfig(load_in_4bit=True, bnb_4bit_compute_dtype=torch.bfloat16)
    11 nf4_config = BitsAndBytesConfig(
    12     load_in_4bit=True,
    13     bnb_4bit_quant_type="nf4",
    14 )
    15 model_nf4 = AutoModelForCausalLM.from_pretrained(model_name, quantization_config=nf4_config)
```

Quantization Speed

7B LLM -> 67s

```
time take to forward 1 without quantization is: 67.70194411277771
```

4bit quantization -> 5.5s

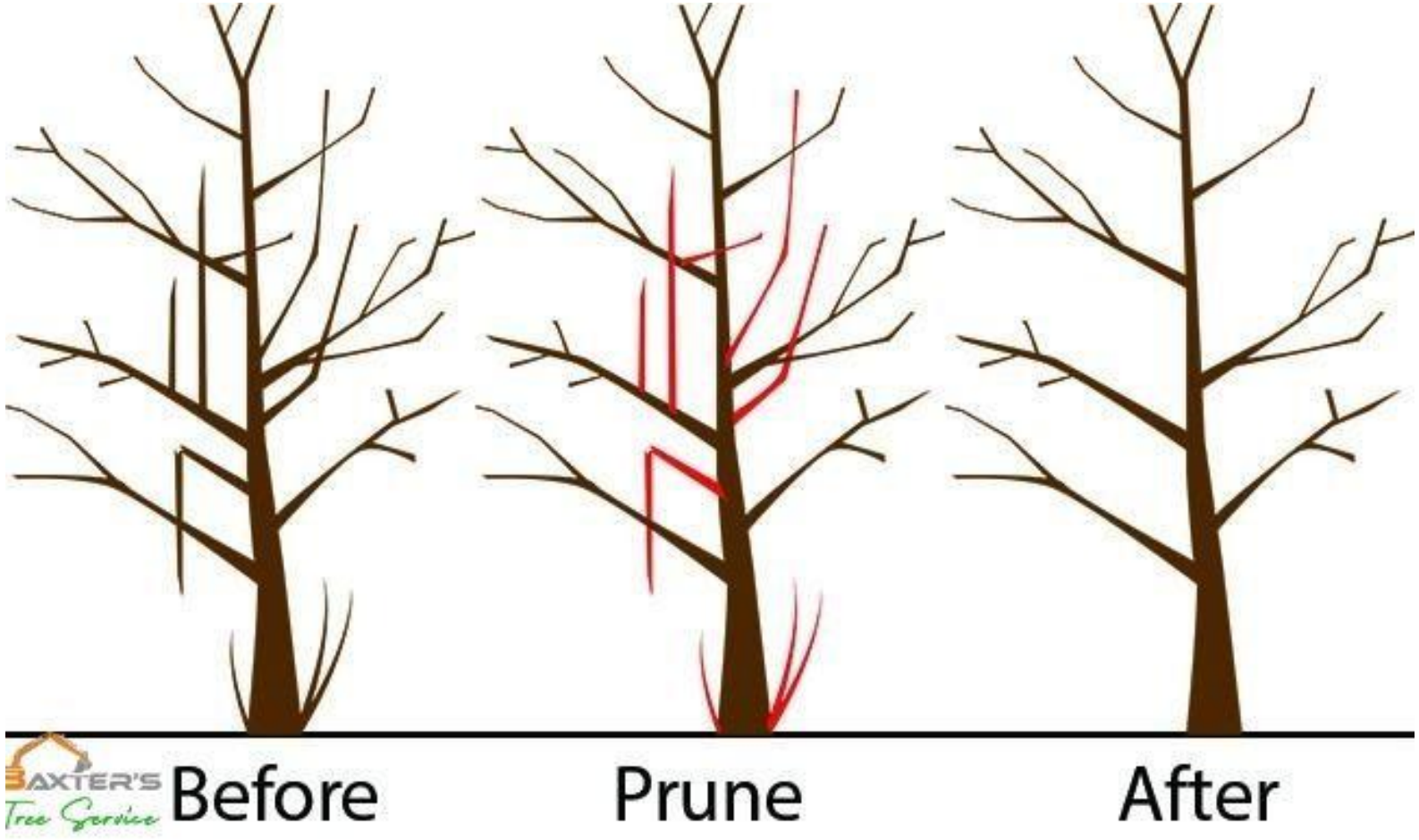
```
Special tokens have been added in the vocabulary, make sure the associated word embeddings are fine-tuned or trained.
time take to forward 1 without quantization is: 5.531926870346069
```

AI VIETNAM
All-in-One Course

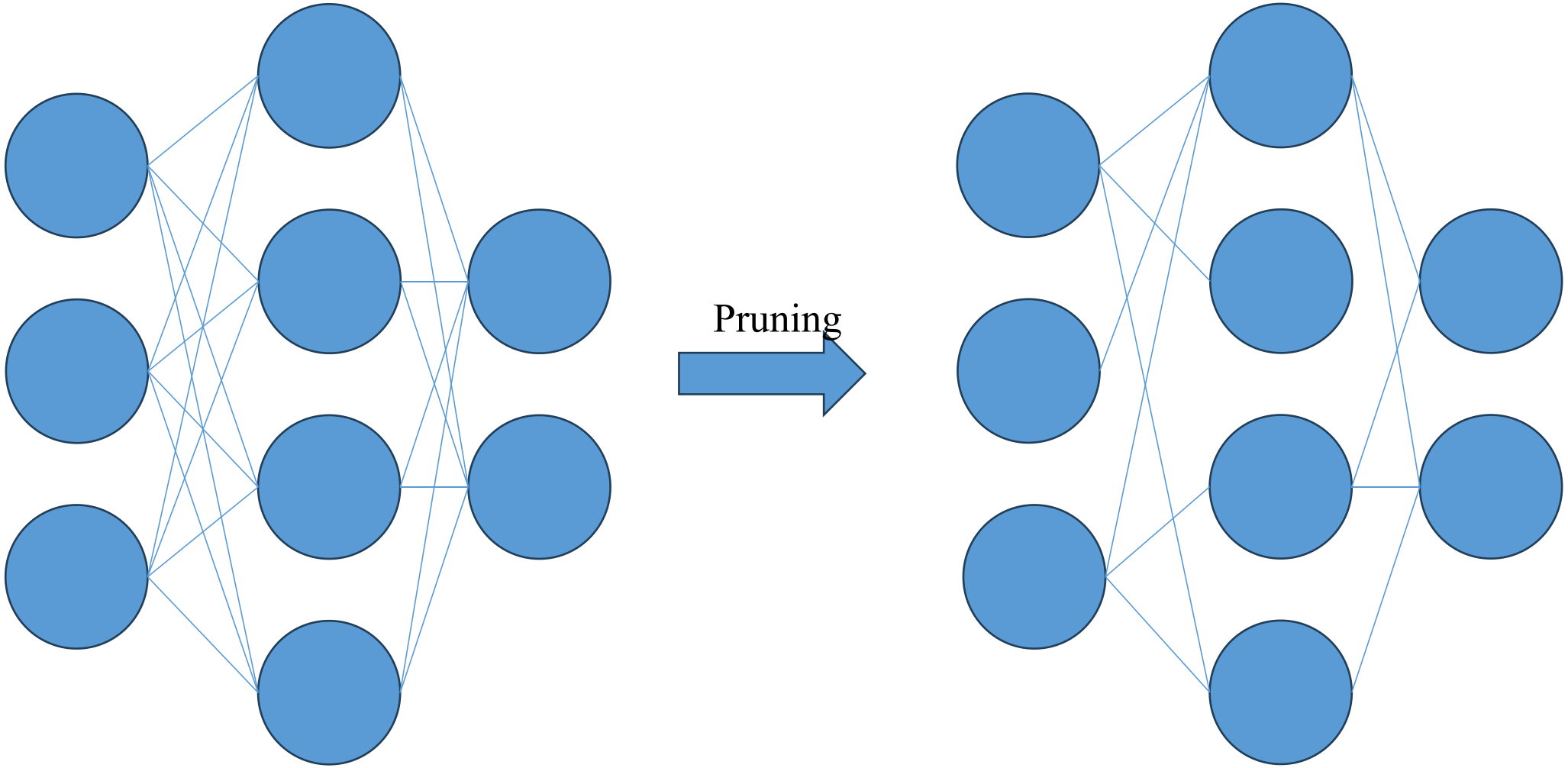
Pruning

- What is Pruning?
- Unstructured vs. Structured pruning
- When to prune?
- Lottery Ticket Hypothesis

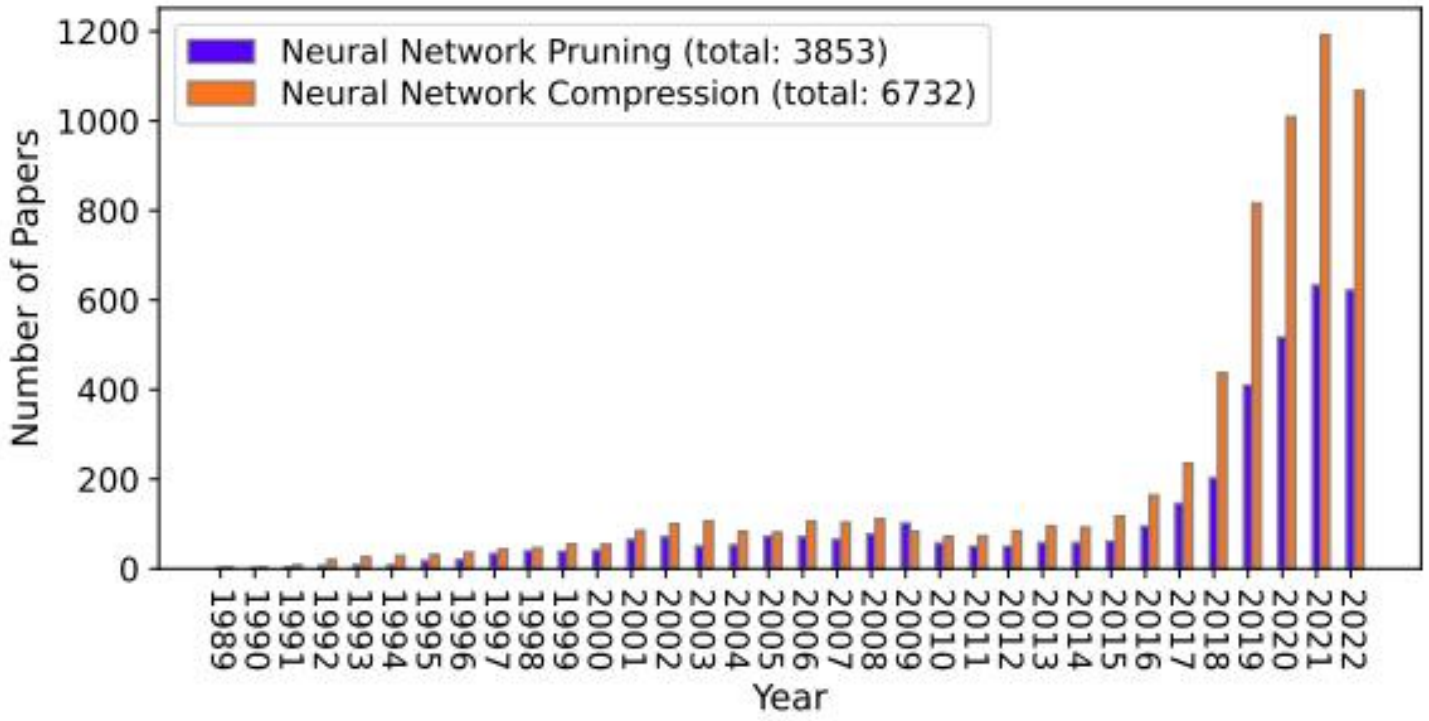
What is pruning



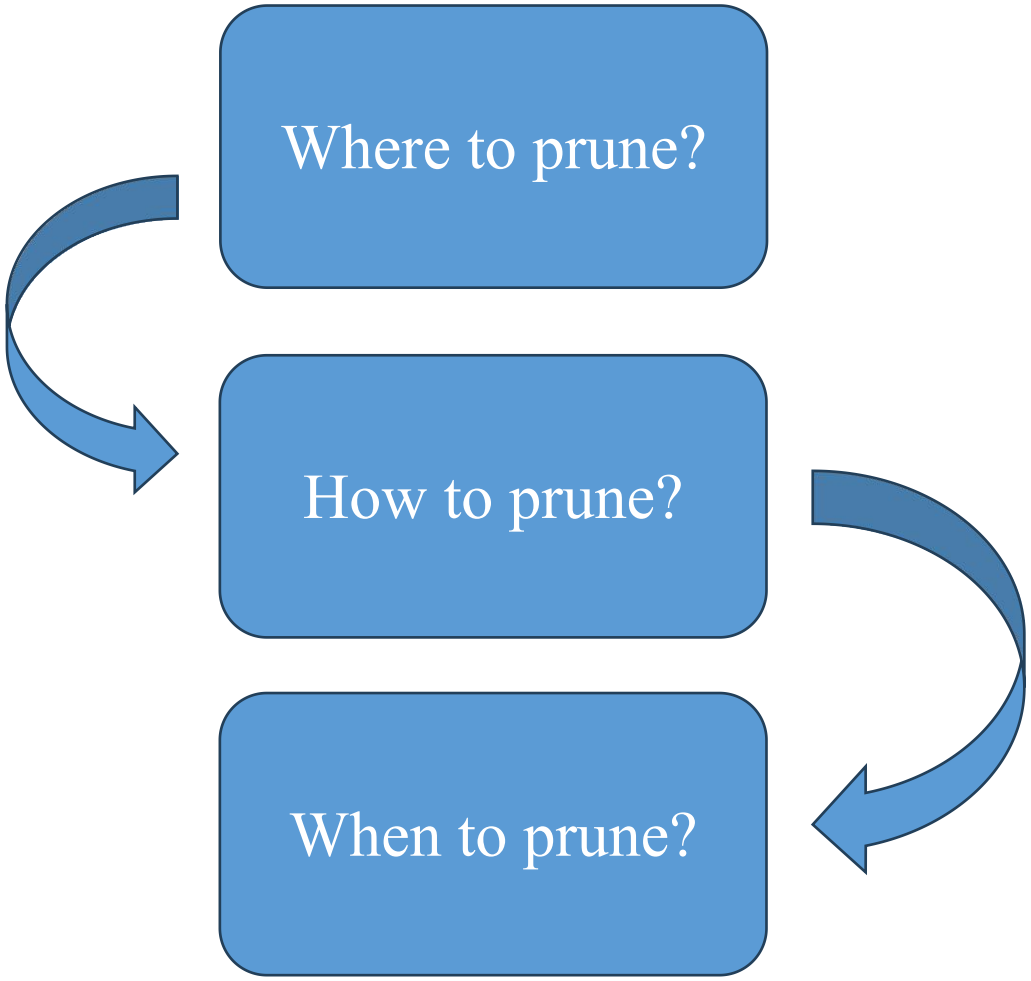
What is pruning



Research on Pruning



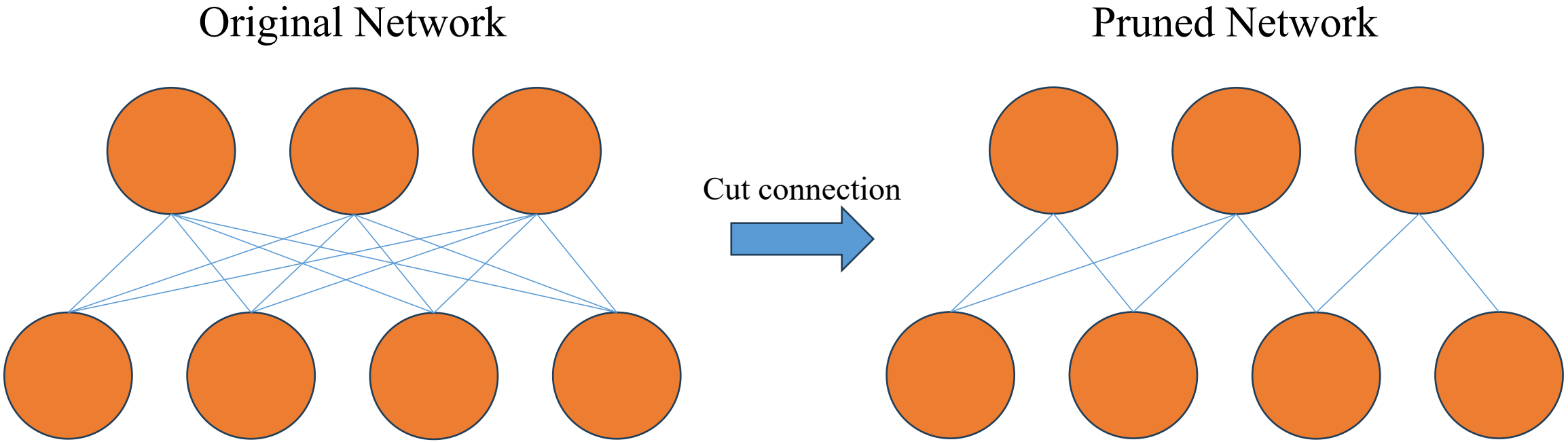
Main Questions



Pruning

- What is Pruning?
- **Unstructured vs. Structured pruning**
- When to prune?
- Lottery Ticket Hypothesis

Unstructured Pruning



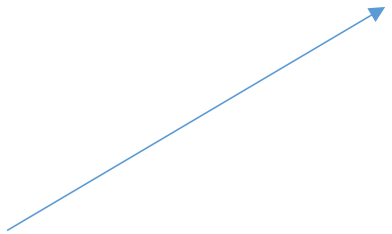
What does this mean?

Unstructured Pruning

1
3
4

0.4	0.8	0.9	6.2
-8.7	-4.3	2.5	5.1
6.7	9.9	1.0	2.4

1.1
27.5
12.4
31.1



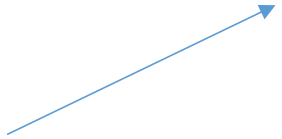
This is connection!

Unstructured Pruning

1
3
4

0.4	0.8	0.9	0
-8.7	-4.3	2.5	5.1
6.7	0	0	2.4

1.1
-12.1
8.4
24.9



This is connection!

Unstructured Pruning

0.4	0.8	0.9	6.2
-8.7	-4.3	2.5	5.1
6.7	9.9	1.0	2.4

*

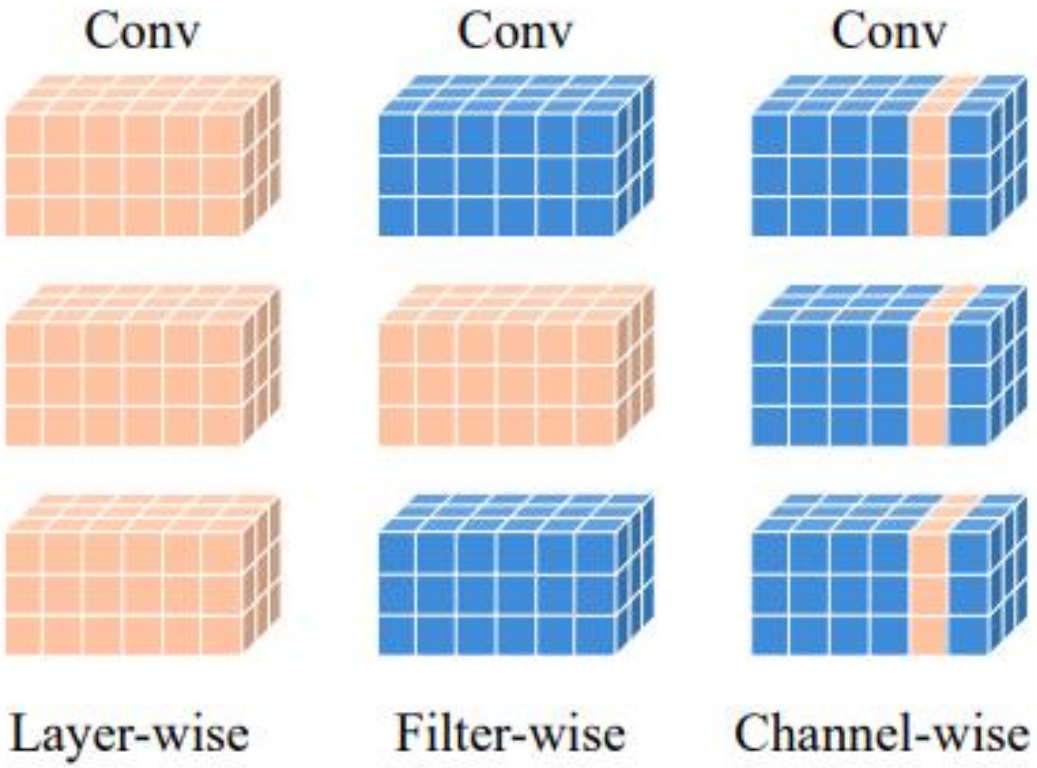
1	1	1	0
1	1	1	1
1	0	0	1

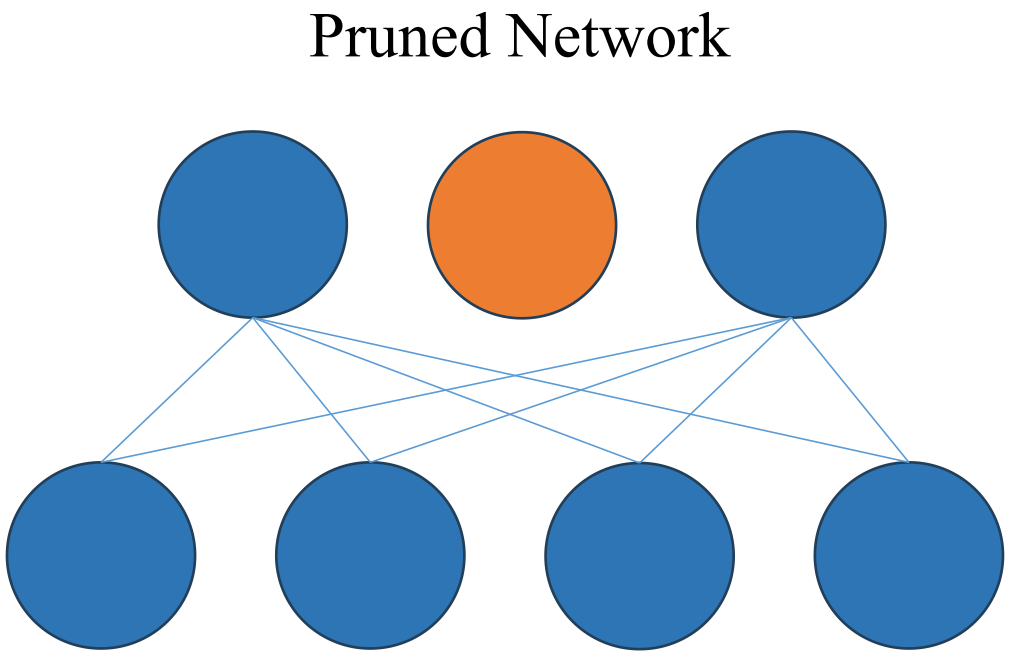
=

0.4	0.8	0.9	0
-8.7	-4.3	2.5	5.1
6.7	0	0	2.4

Need for Special Software!!!

Structured Pruning

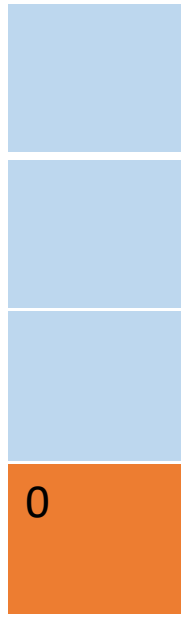


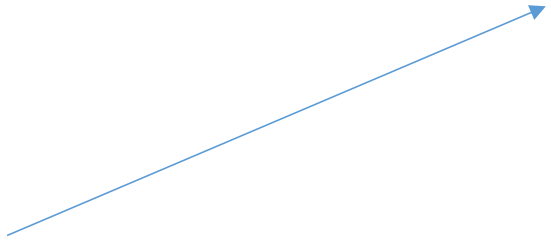


Structured Pruning

1
3
4

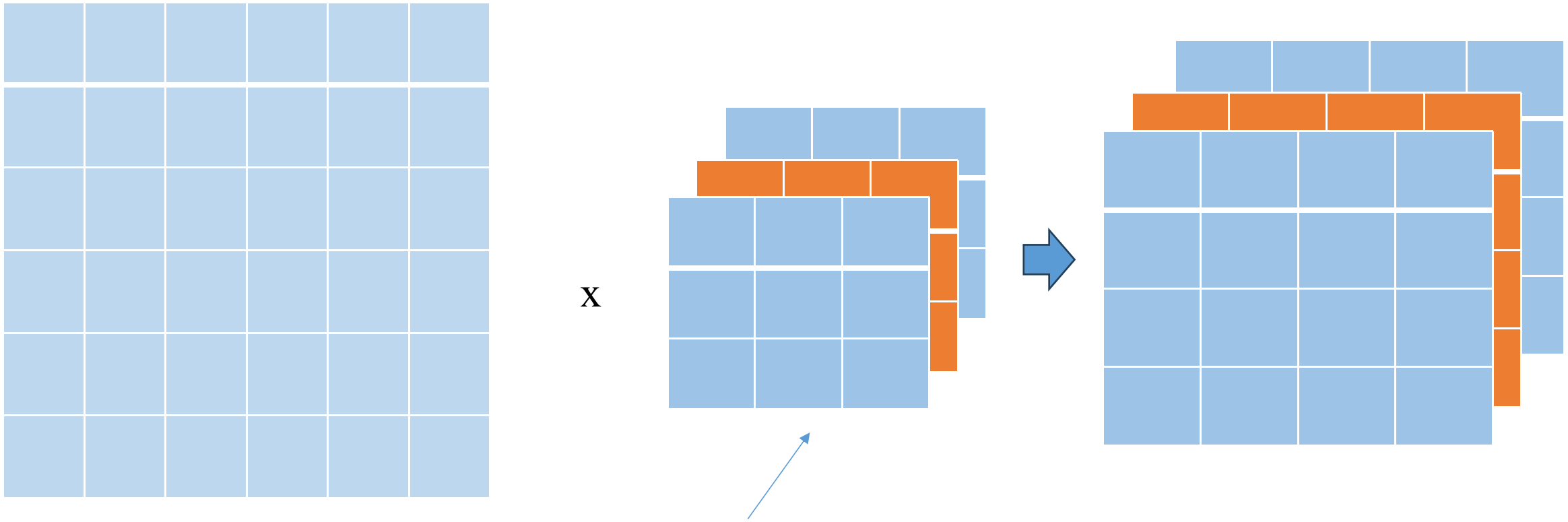
0.4	0.8	0.9	0
-8.7	-4.3	2.5	0
6.7	9.9	1.0	0





Prune a layer!

Structured Pruning



Structured vs. Unstructured

	Unstructured	Structured
High sparsity with minor accuracy drop	Yes	Hard
Speedup w/o specific hardware	Hard	Yes
Speedup w/o specific software	Hard	Yes
Really compressed with significant acceleration	Hard	Yes
Structure coupling	No	Yes

Pruning

- What is Pruning?
- Unstructured vs. Structured pruning
- How & when to prune?
- Lottery Ticket Hypothesis

How To Prune?

Magnitude-based Pruning

1.2	2.4	-5.6
3.2	-4.1	1.0
0.8	4.4	-2.2

Prune Ratio: 40%

0.8	1.0	1.2	2.2	2.4	3.2	4.1	4.4	5.6

How To Prune?

Magnitude-based Pruning

1.2	2.4	-5.6
3.2	-4.1	1.0
0.8	4.4	-2.2

1.2	2.4	-5.6
3.2	-4.1	1.0
0.8	4.4	-2.2

Prune Ratio: 40%

0.8	1.0	1.2	2.2	2.4	3.2	4.1	4.4	5.6

How To Prune?

Magnitude-based Pruning

1.2	2.4	-5.6
3.2	-4.1	1.0
0.8	4.4	-2.2

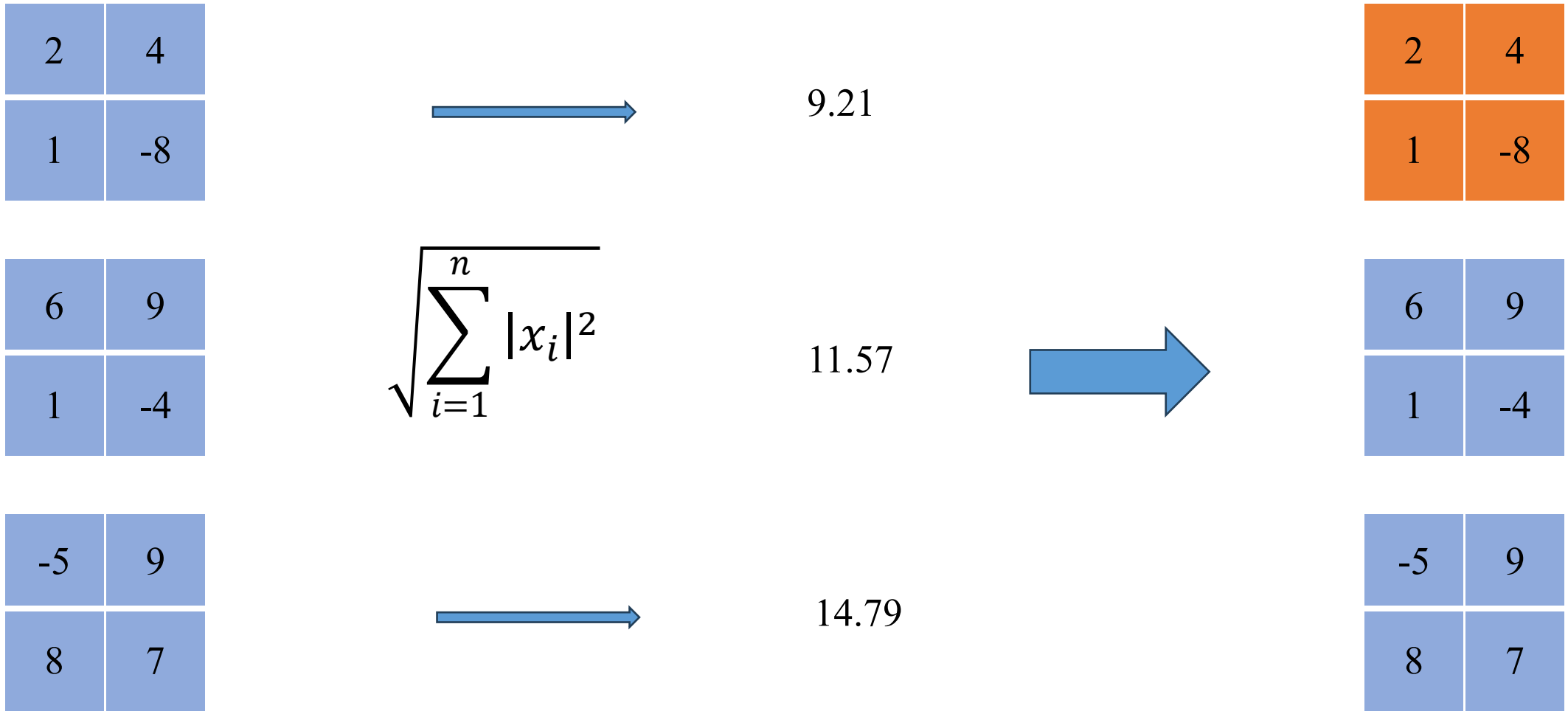
0.0	2.4	-5.6
3.2	-4.1	0.0
0.0	4.4	0.0

Prune Ratio: 40%

0.8	1.0	1.2	2.2	2.4	3.2	4.1	4.4	5.6

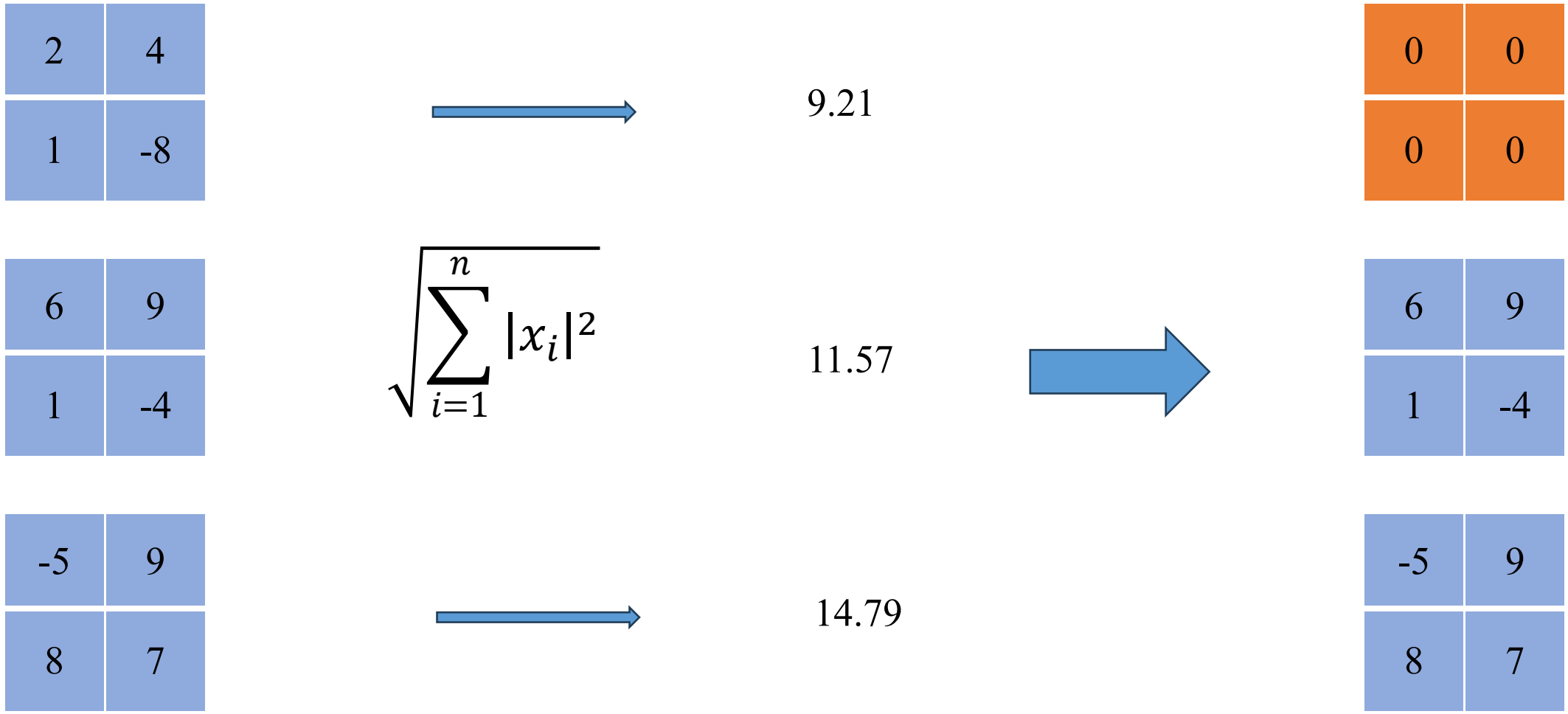
How To Prune?

L2 Norm Pruning

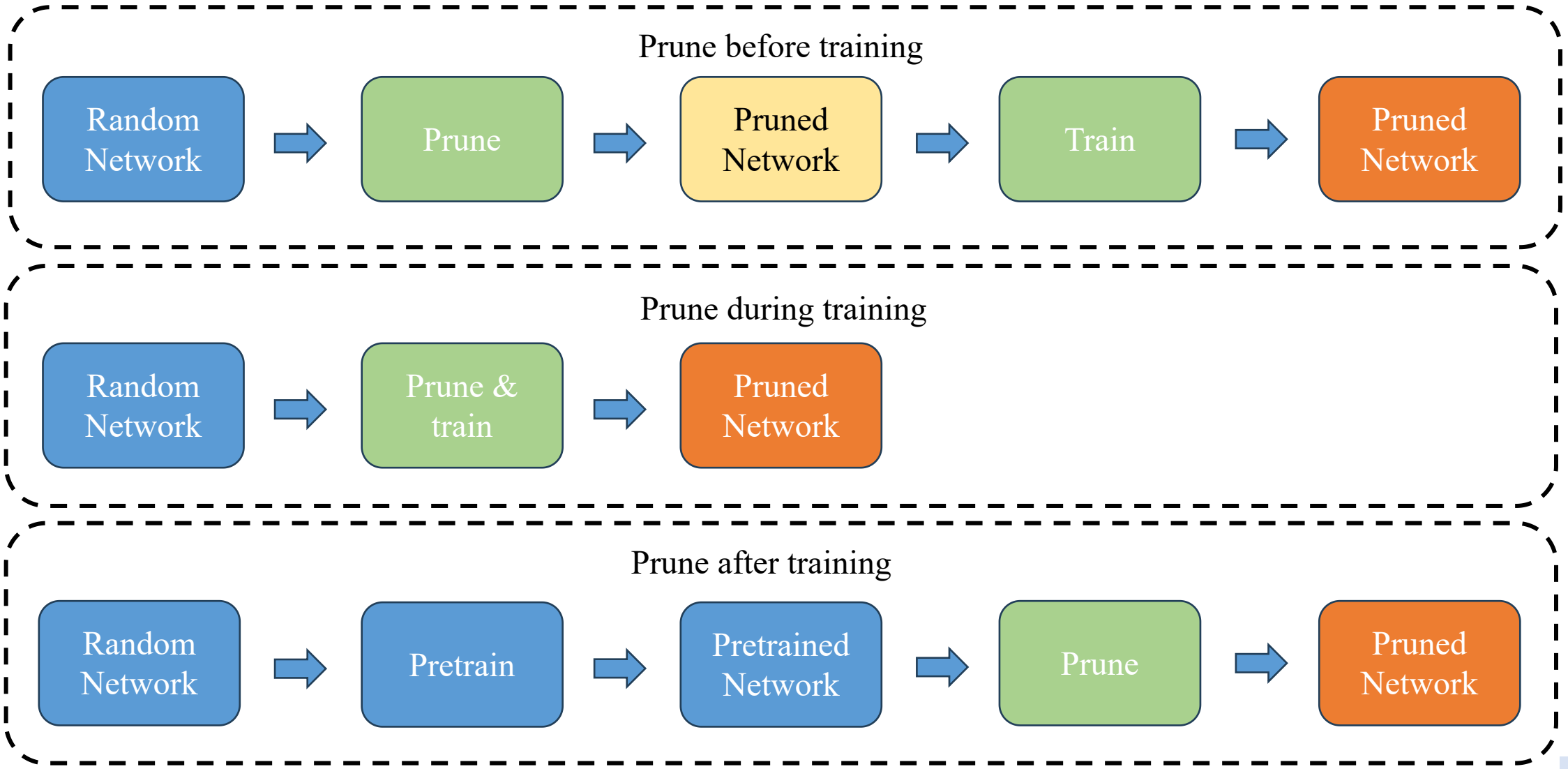


How To Prune?

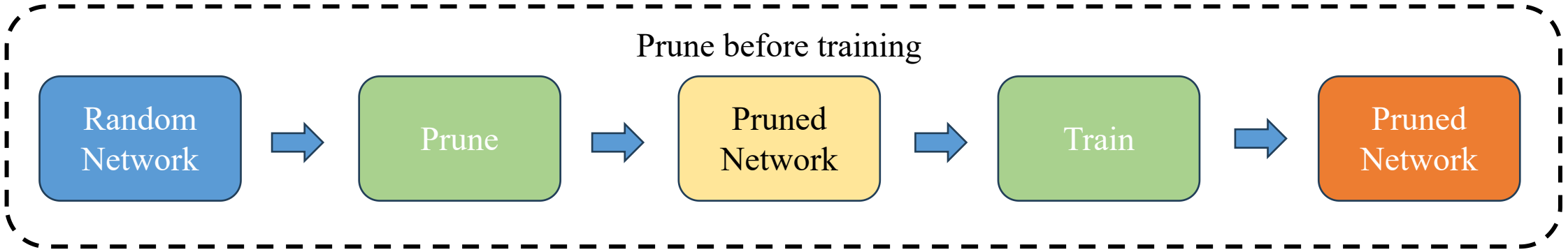
L2 Norm Pruning

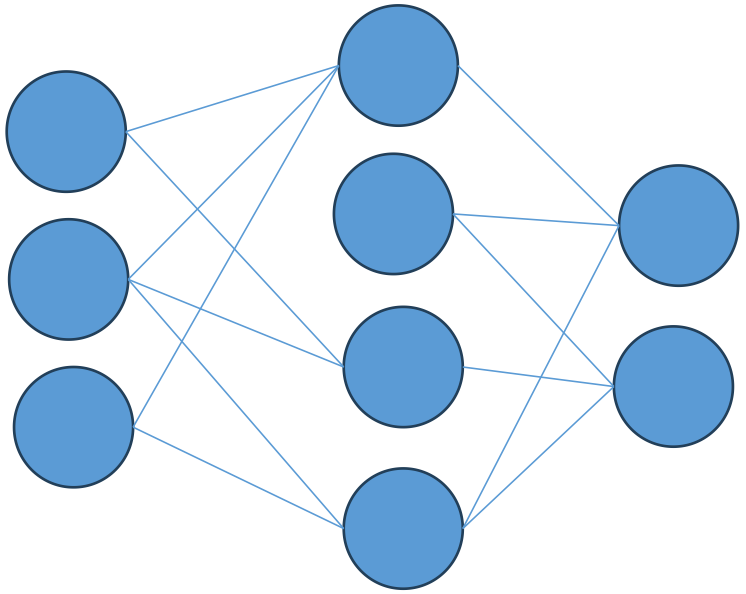


When to prune?



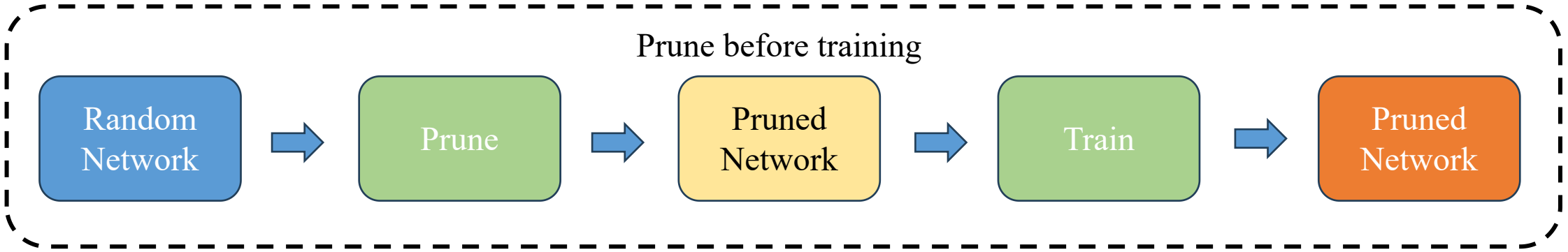
Prune Before Training

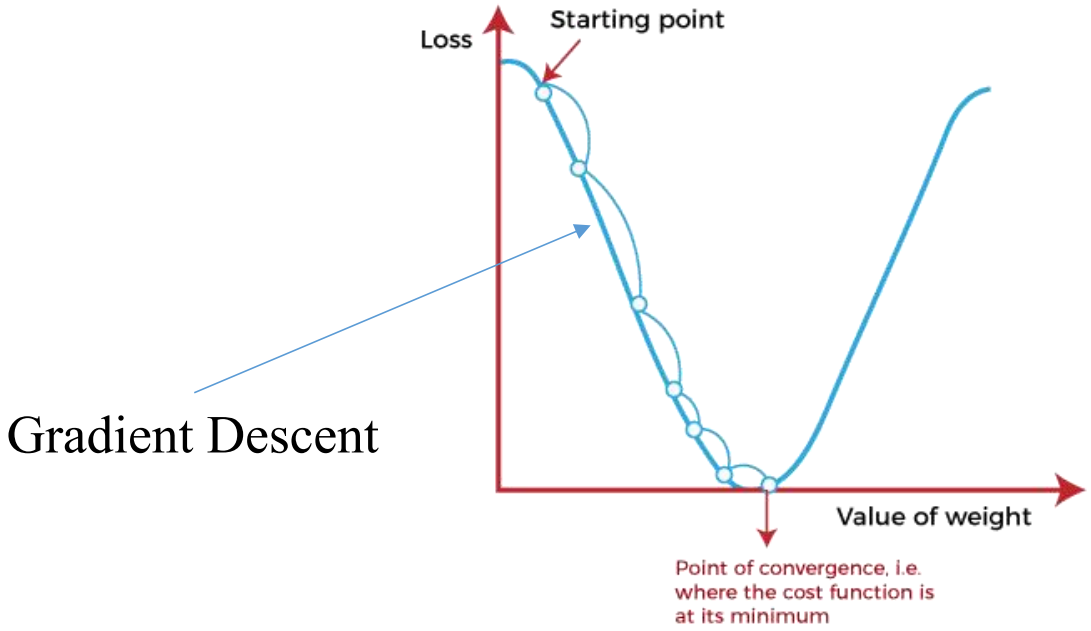


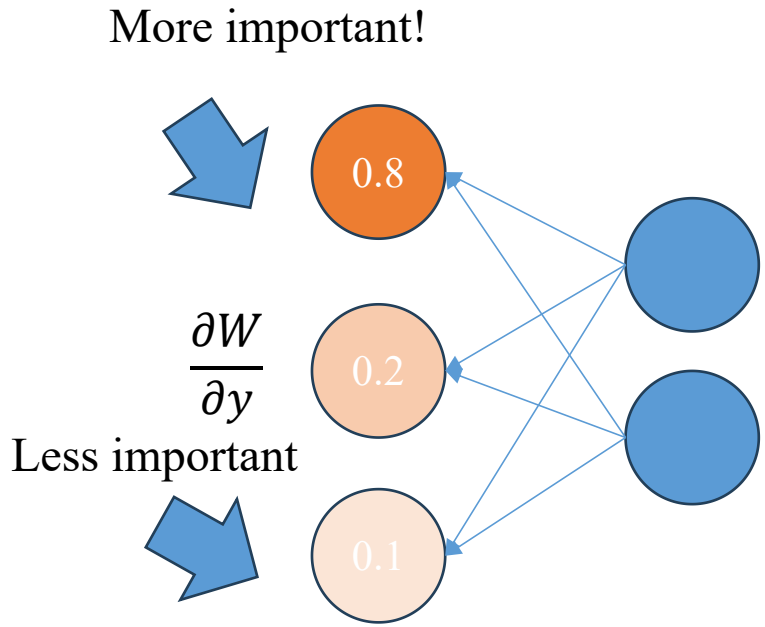


Simplest Method:
Random Pruning!

Prune Before Training







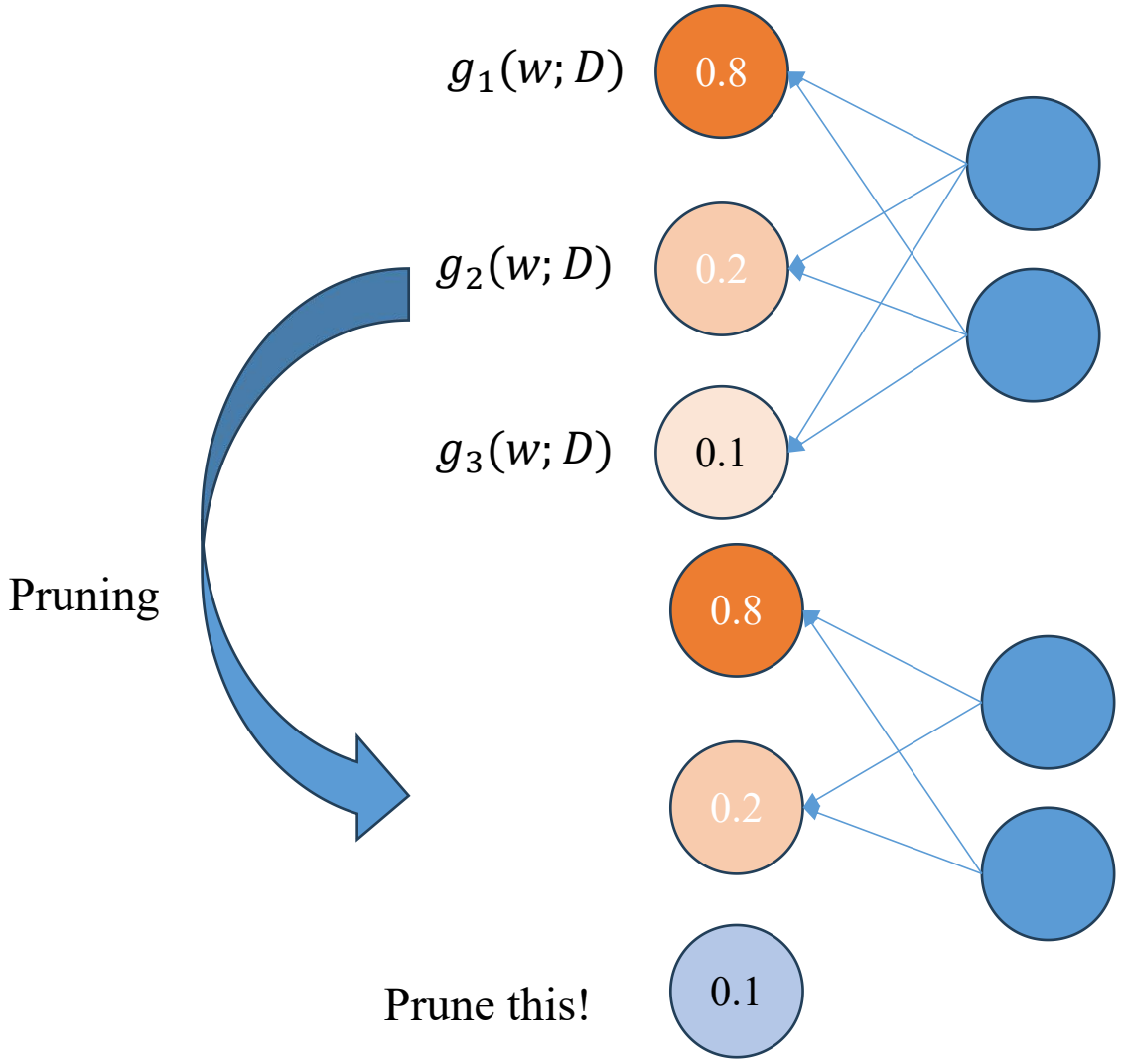
SNIP

$$s_j = \frac{|g_j(w; D)|}{\sum_{k=1}^{m} |g_k(w; D)|}$$

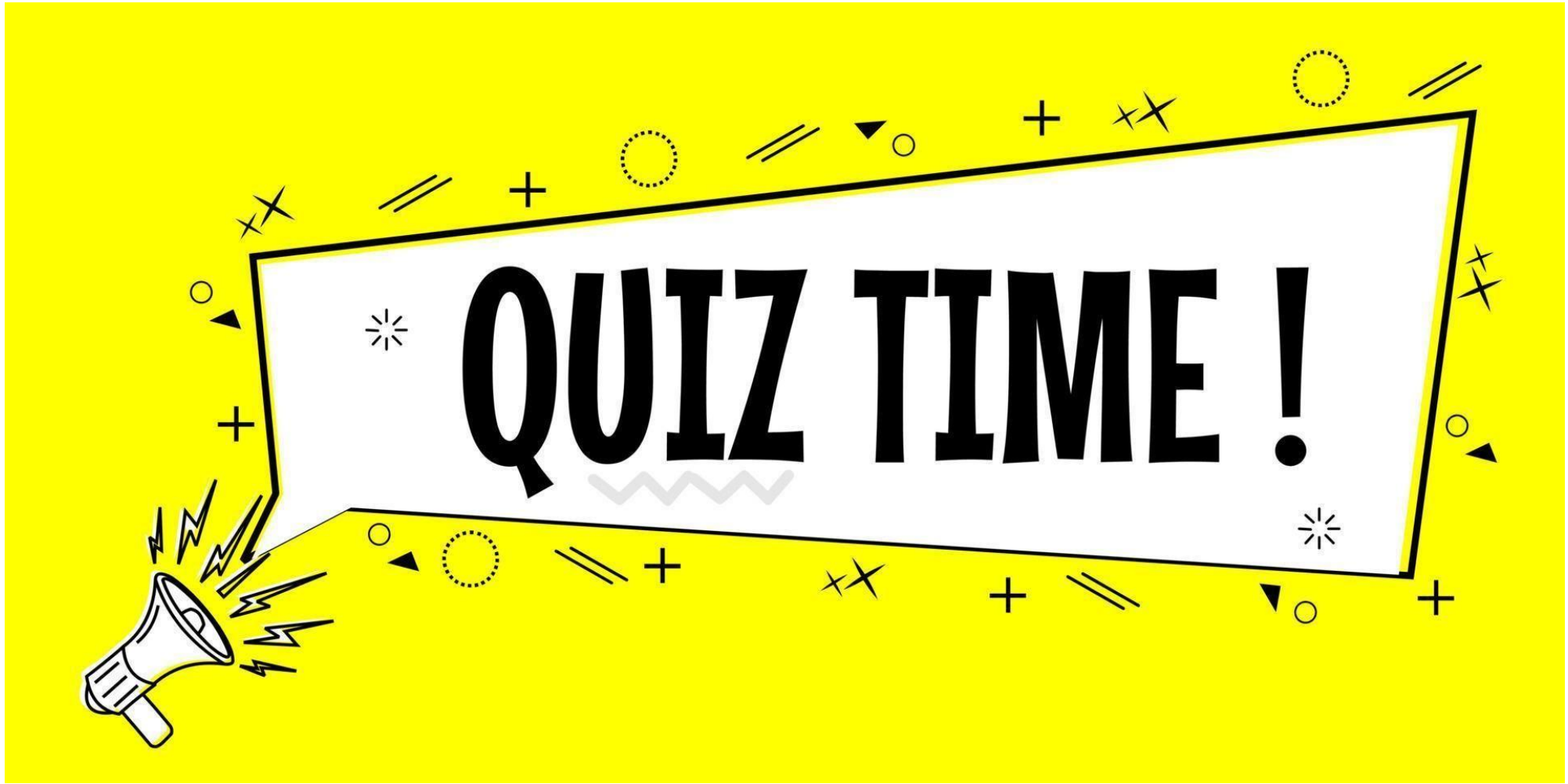
$$S_1 = \frac{0.8}{0.8 + 0.2 + 0.1} = 0.73$$

$$S_1 = \frac{0.2}{0.8 + 0.2 + 0.1} = 0.18$$

$$S_1 = \frac{0.1}{0.8 + 0.2 + 0.1} = ?$$



QUIZ TIME!!!



QUIZ TIME!!!

Câu 1: Mục tiêu của việc pruning trong mạng nơ-ron là gì?
A) Tăng khả năng giải thích của mô hình
B) Giảm kích thước model và nguồn lực tính toán
C) Cải thiện dữ liệu huấn luyện
D) Tối đa hóa số lớp

Câu 2: Unstructured pruning nhắm vào mục tiêu nào sau đây?
A) Toàn bộ các lớp
B) Kiến trúc cụ thể
C) Các trọng số trong một lớp
D) Đặc điểm đầu vào của dữ liệu

QUIZ TIME!!!

Câu 3: Structured pruning được đặc trưng bởi việc loại bỏ:
A) Trọng số ngẫu nhiên
B) Toàn bộ nơ-ron hoặc kênh
C) Điểm dữ liệu đầu vào
D) Tốc độ học

Câu 4: Magnitude Pruning thường liên quan đến:
A) Tỉa các trọng số gần giá trị trung bình nhất
B) Tỉa các trọng số được cập nhật gần đây nhất
C) Tỉa trọng số dựa trên mã màu
D) Tỉa các trọng số có độ lớn nhỏ nhất

QUIZ TIME!!!

Bài 5: Thách thức chính trong việc áp dụng các phương pháp pruning là:

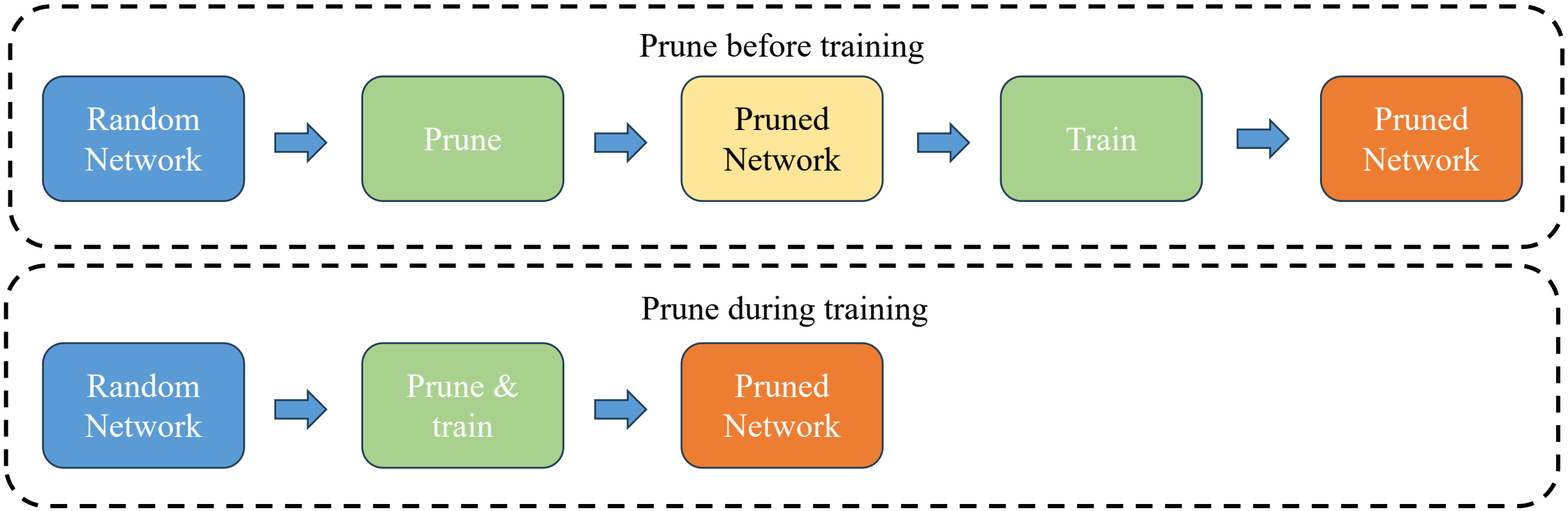
A) Làm cho mô hình lớn hơn

B) Duy trì hiệu suất mô hình trong khi giảm độ phức tạp

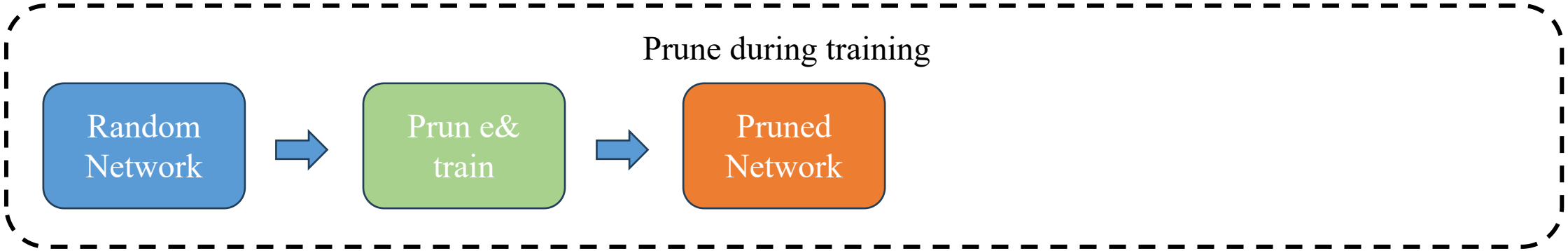
C) Tăng số lượng tham số

D) Đơn giản hóa kiến trúc mô hình một cách không cần thiết

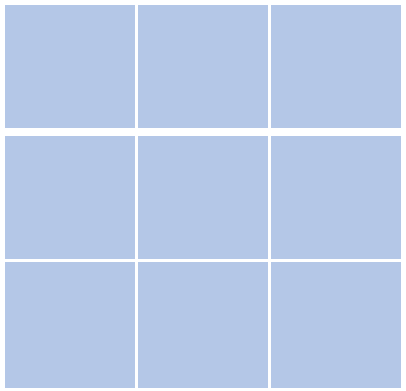
When to prune?

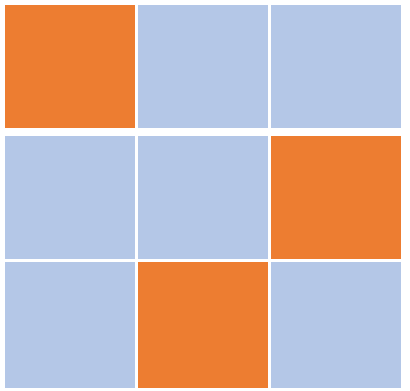


Prune During Training



Update Weights + Mask





Prune During Training

Neural Network

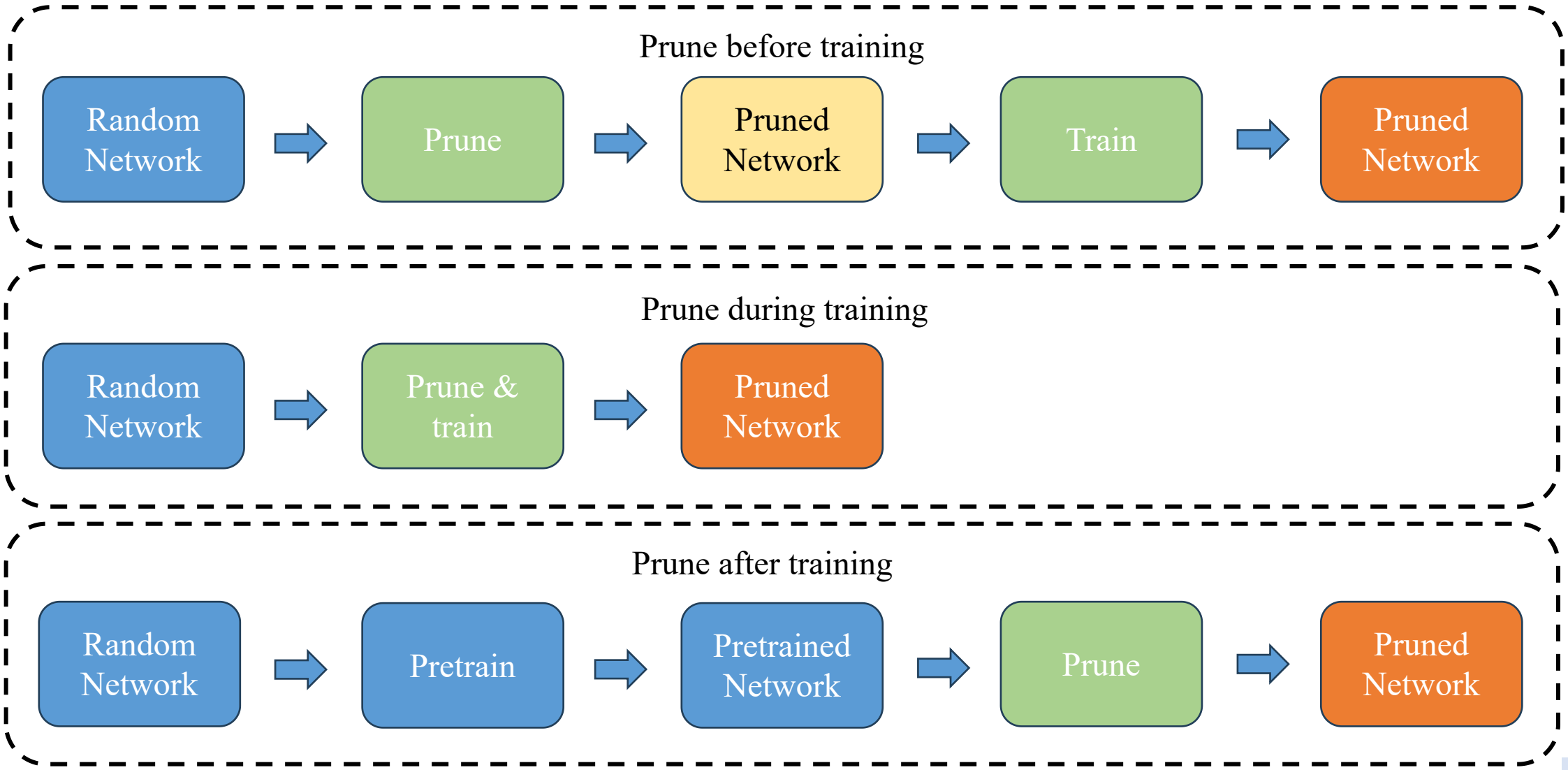
Prune x% for all layers

Other Methods

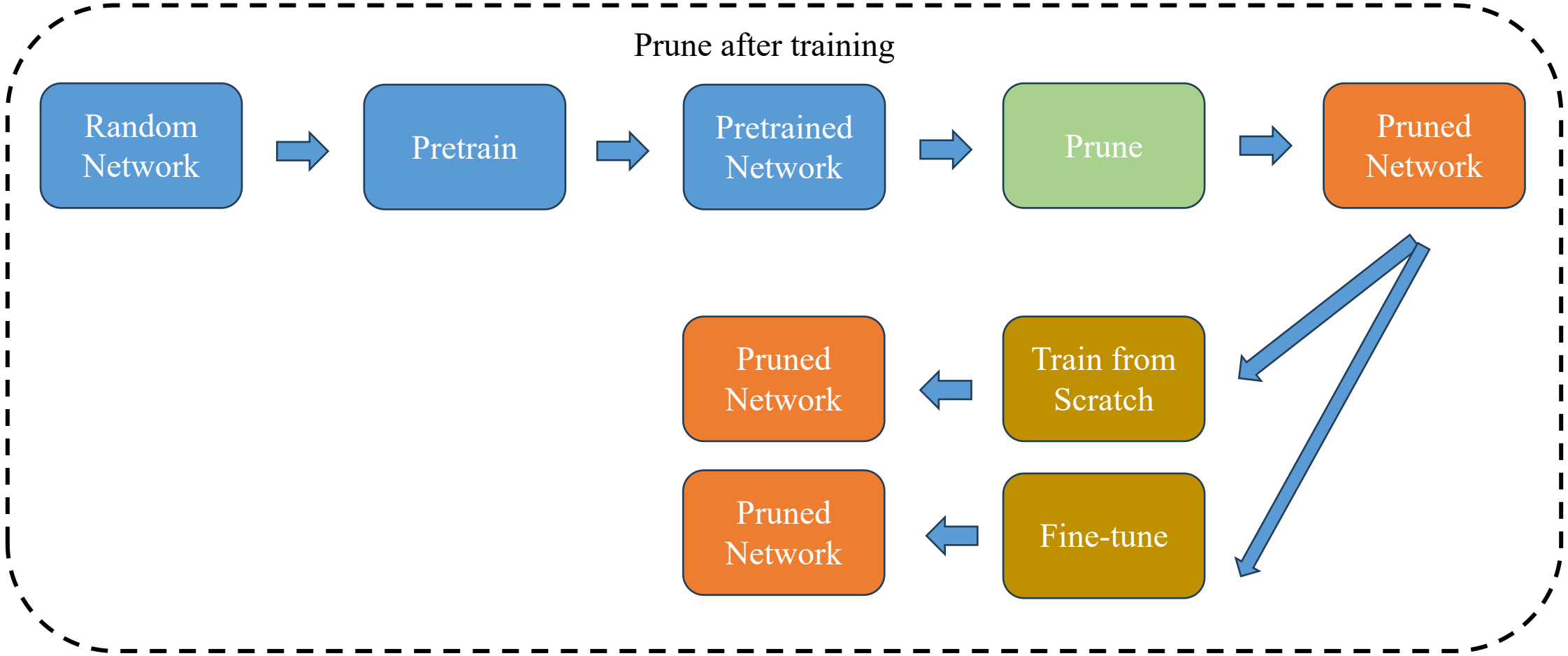
Learn the suitable % for each layer

Prune During Training

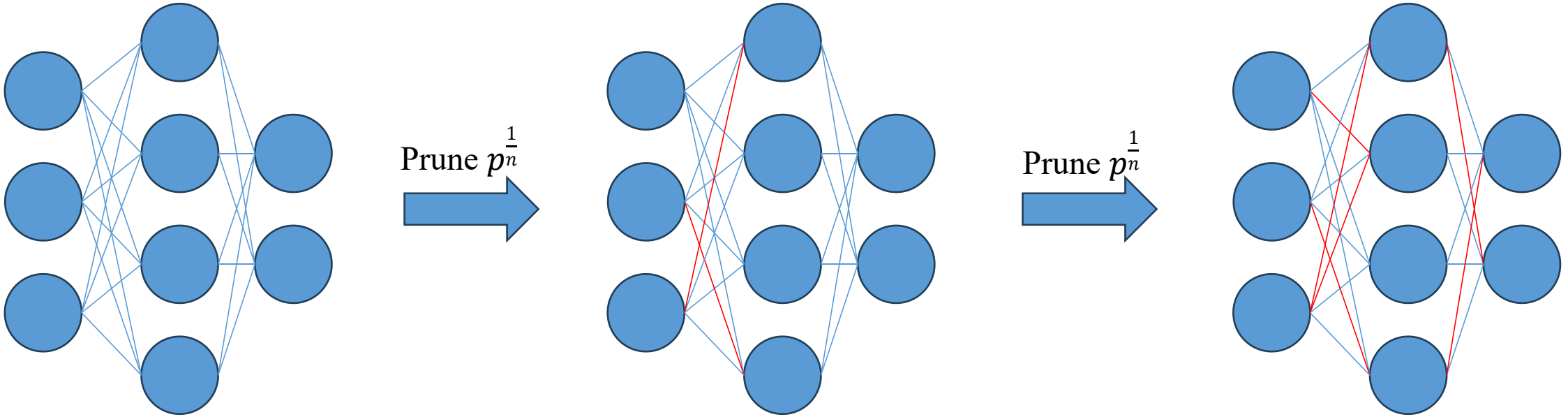
When to prune?



When to prune?

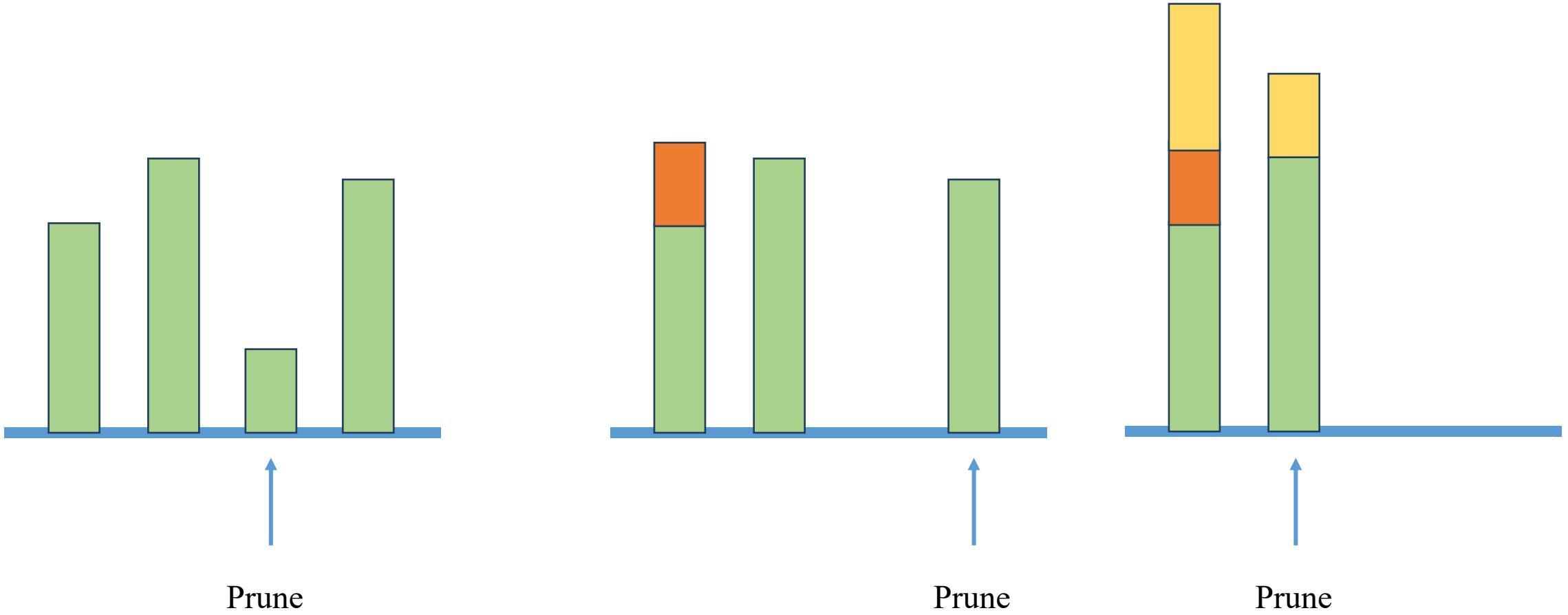


Iterative Magnitude Pruning



Iterative Pruning

Iterative Magnitude Pruning



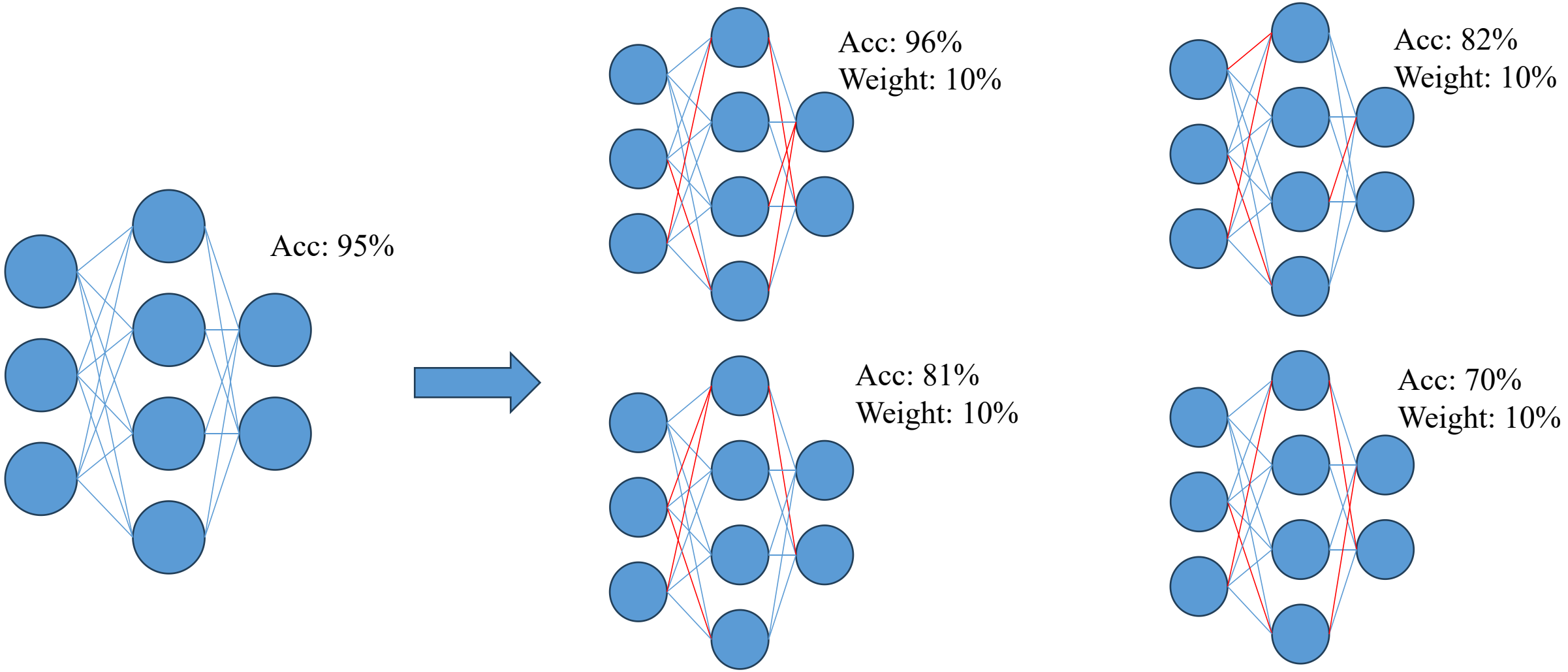
AI VIETNAM
All-in-One Course

Pruning

- What is Pruning?
- Unstructured vs. Structured pruning
- How & When to prune?
- Lottery Ticket Hypothesis

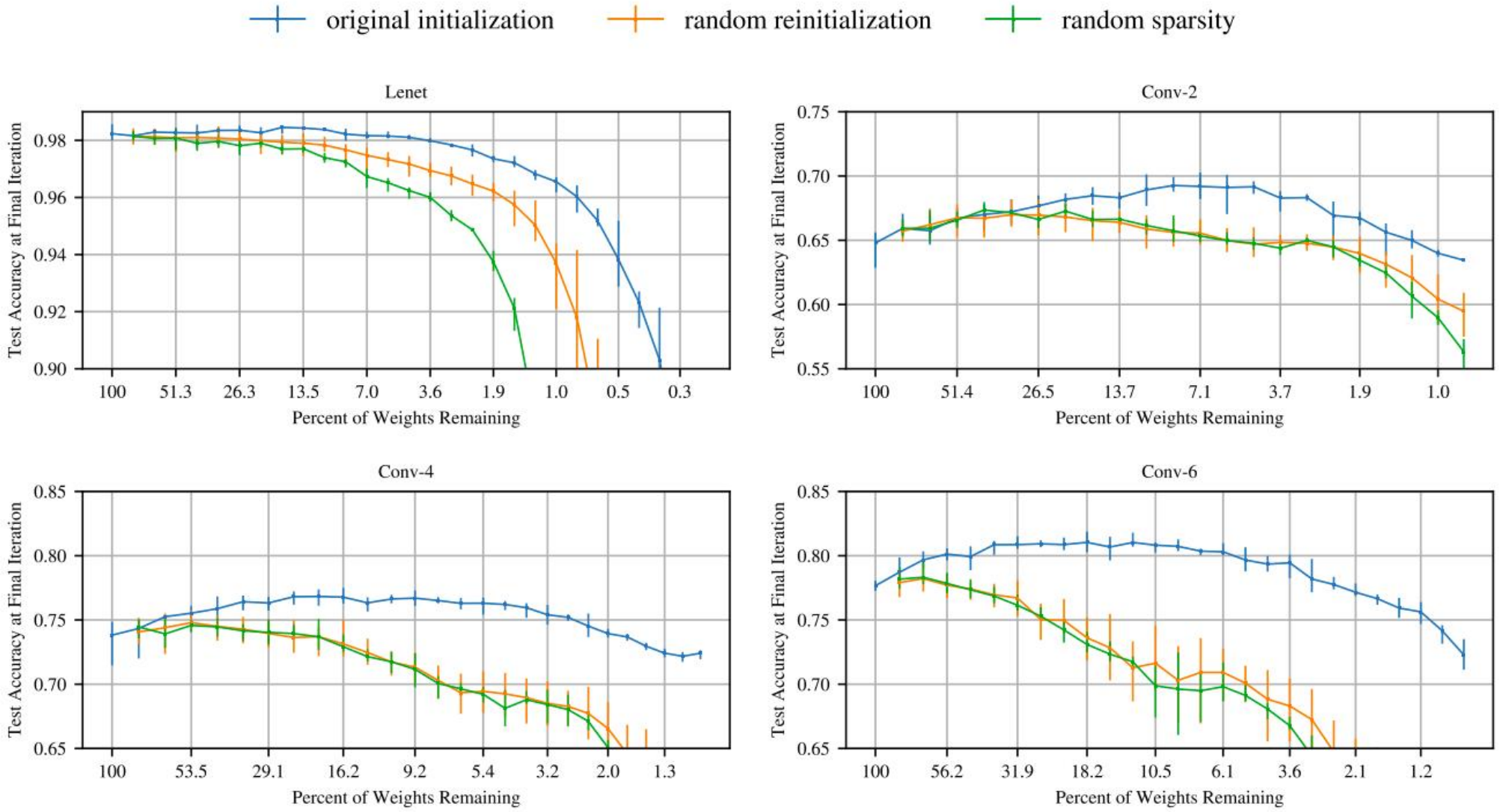
Lottery Ticket Hypothesis

Lottery Ticket Hypothesis

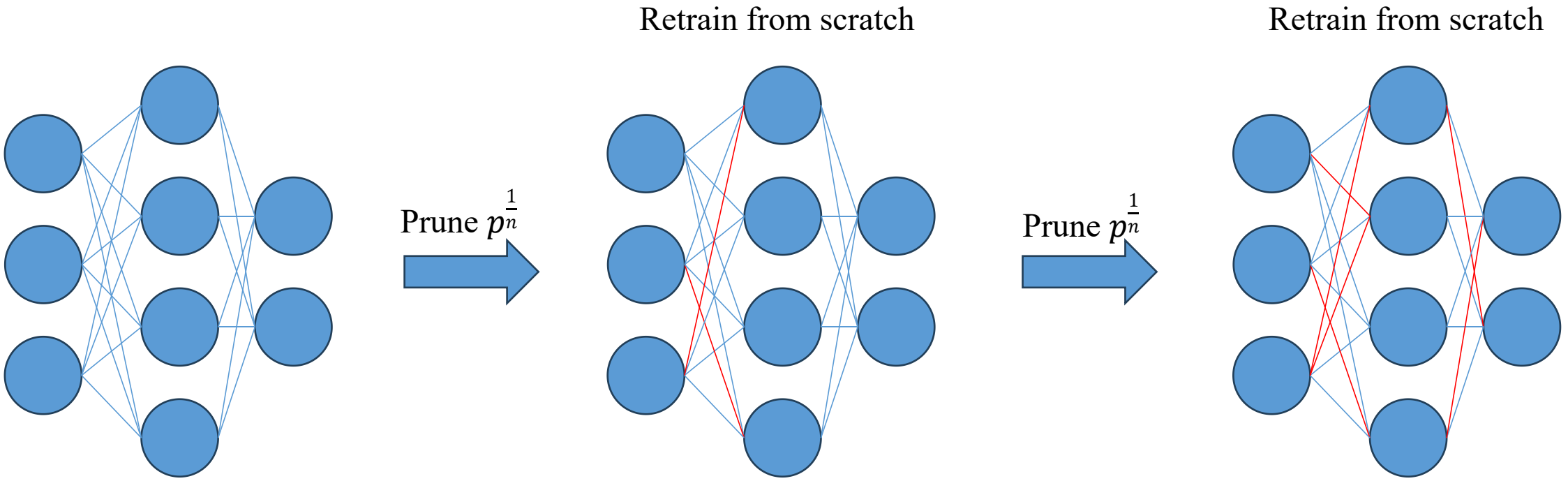


Larger Models ~ Buy more tickets

Lottery Ticket Hypothesis



Iterative Magnitude Pruning



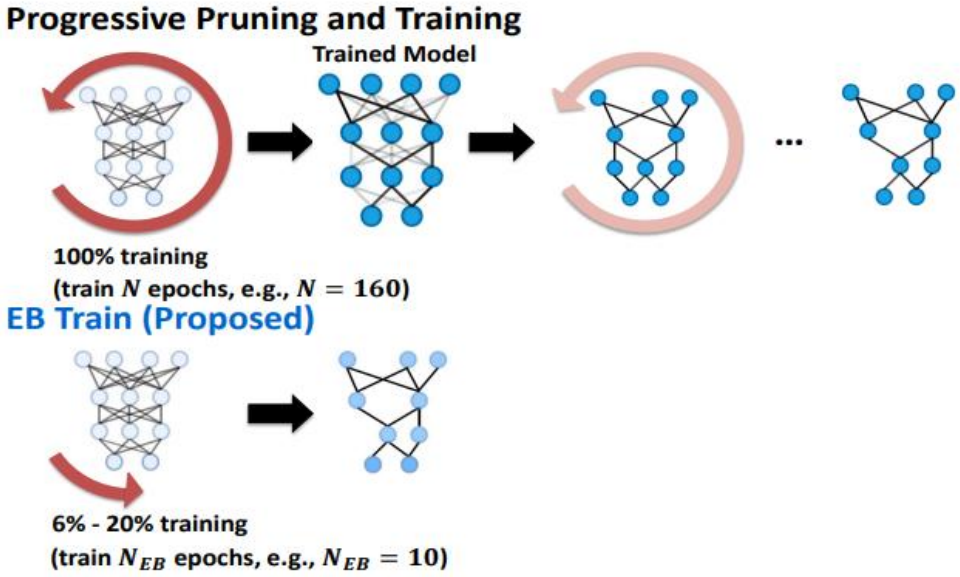
Iterative Pruning

Identify the winning ticket

Step 1: Randomly initialize a network
Step 2: Train the network for j iteration
Step 3: Prune p% of the parameters
Step 4: Reset the parameters to their original values
Step 5: Retrain from scratch
Step 6: Loop
Step 7: Achieve Winning Ticket

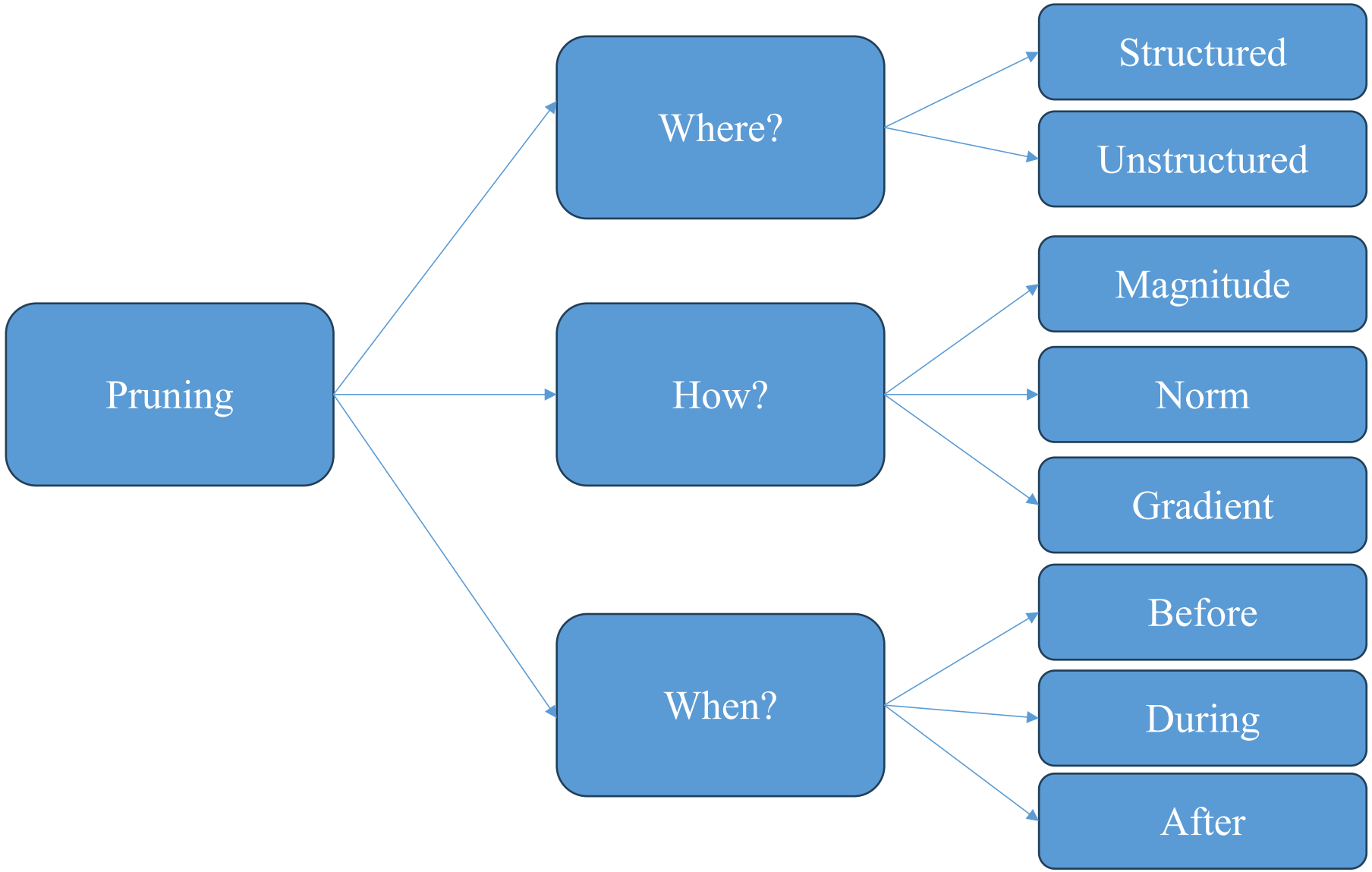
Further research

Early Bird Ticket: 1909.11957.pdf (arxiv.org)



Mathematical Proof: 2002.00585.pdf (arxiv.org)

Summary


Pruning
Where?
How?
When?
Structured
Unstructured
Magnitude
Norm
Gradient
Before
During
After

Pruning