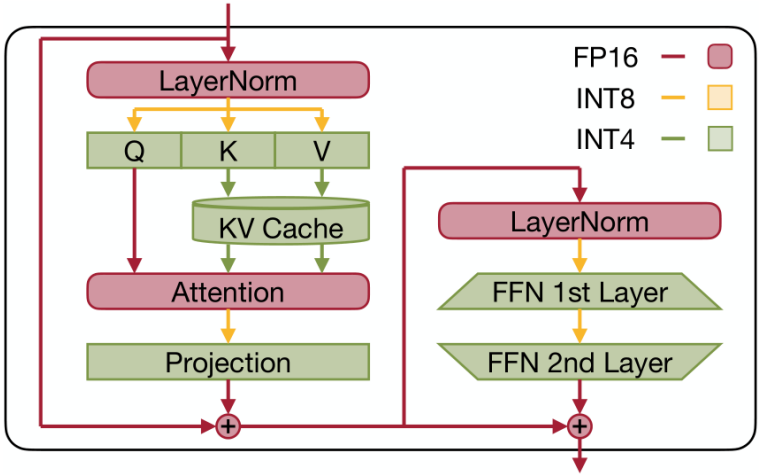
On-Device AI 실습: Pruning for LLM

Dongkun Shin
Intelligent Embedded Systems Lab.
Sungkyunkwan University

Post-Training Pruning for LLMs

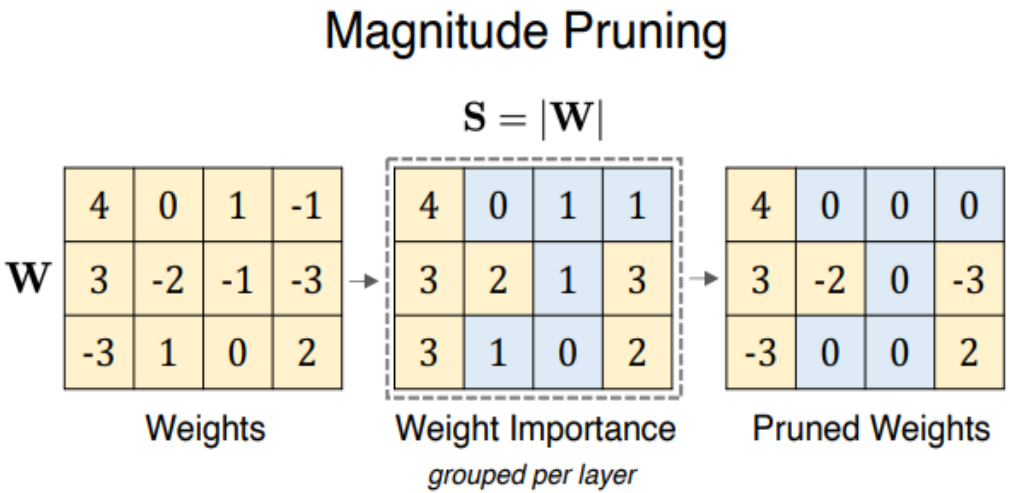
- LLM Pruning 시 pruning 대상은 Multi-head attention의 projection layer (FC) 와 FFN의 FC layer들임.



LLM Architecture

[실습 1] Magnitude-based Pruning 구현

- Weight의 magnitude 만을 이용하여 pruning 수행



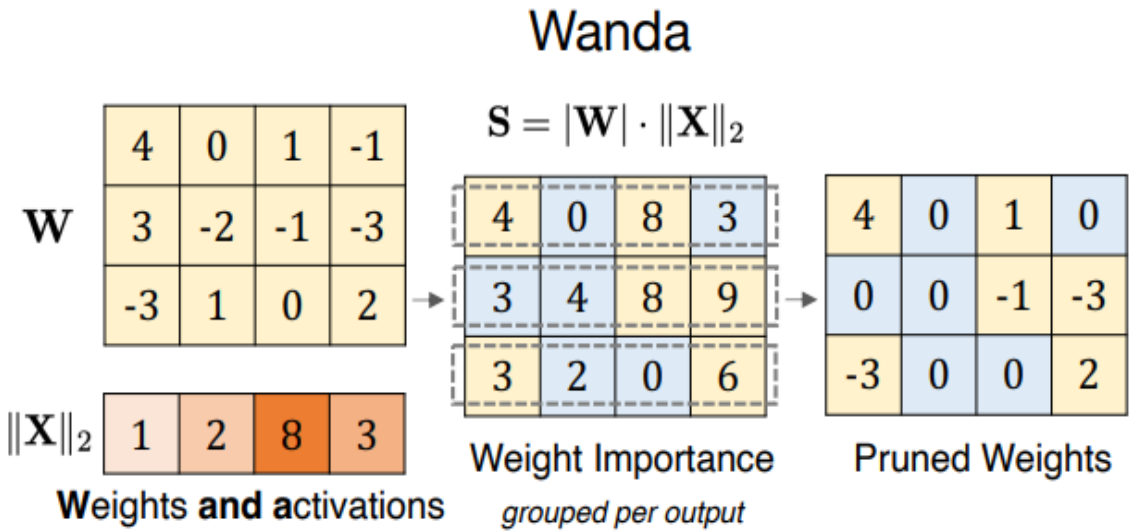
[실습 1] Magnitude-based Pruning


```
##################### YOUR CODE STARTS HERE #####################
num_elements = W.numel()
num_zeros = round(num_elements * sparsity)
importance = torch.abs(W)
threshold = torch.kthvalue(importance.flatten(), num_zeros)[0]
mask = importance > threshold
###################### YOUR CODE ENDS HERE ######################
```

[실습 2, 3] Wanda Pruning 구현

- Calibration을 통한 feature의 값을 sampling

$$\|\mathbf{X}\|_2 = \sqrt{\sum_i \mathbf{X}_i^2}$$



[실습 2, 3] Answer


```
###################### YOUR CODE STARTS HERE ######################
# activation_norm을 계산하세요.
# x.shape => (hidden_size, batch_size)
activation_norm = torch.norm(x, p=2, dim=1) ** 2
# activation_norm.shape => (hidden_size)
###################### YOUR CODE ENDS HERE ########################
```

```
###################### YOUR CODE STARTS HERE ######################
row, col = W.shape
num_zeros_per_row = round(col * sparsity)
importance = torch.abs(W) * input_feat[n]
threshold = torch.kthvalue(importance, num_zeros_per_row, dim=1)[0]
mask = importance > threshold.reshape(row, 1)
###################### YOUR CODE ENDS HERE ########################
```