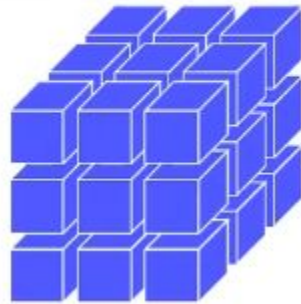


Pytorch 오피스아워

백혜림 조교

주어진 Tensor



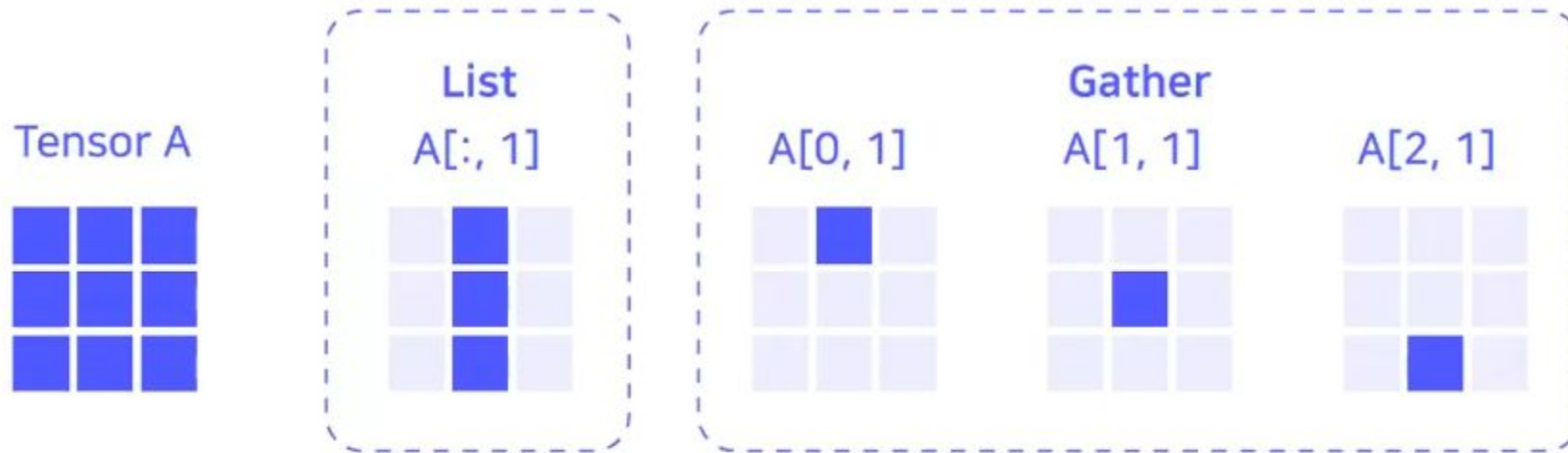
우리가 가져오고 싶은 값들



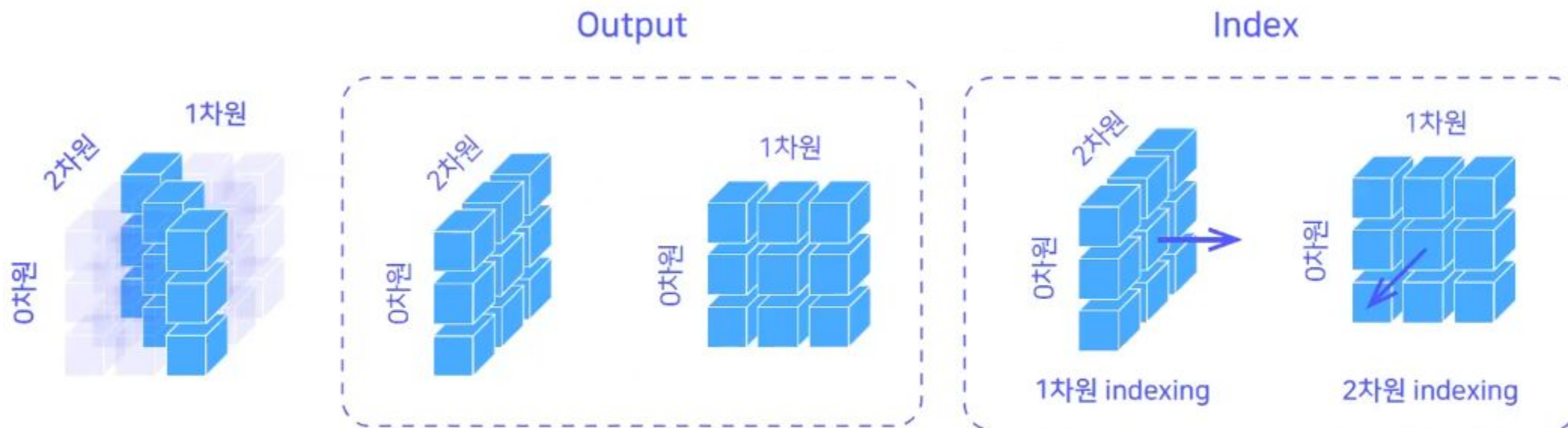


```
out[i][j][k] = input[index[i][j][k]][j][k] # if dim == 0  
out[i][j][k] = input[i][index[i][j][k]][k] # if dim == 1  
out[i][j][k] = input[i][j][index[i][j][k]] # if dim == 2
```

Index Tensor의 크기는 우리가 원하는 Output Tensor의 크기와 같다
Gather의 Index는 각 원소마다 하나씩 값을 지정해주어야 한다

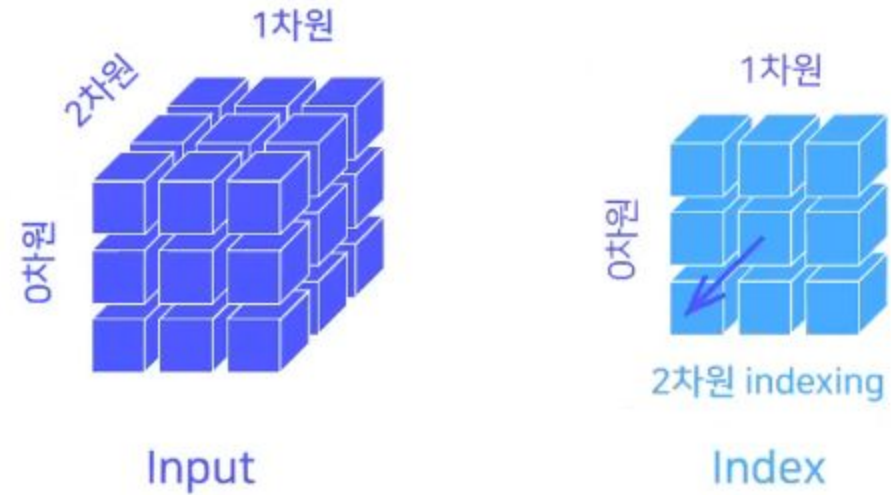


List에서 indexing을 하던 것처럼 한번에 여러 요소를 선택할 수가 없다!
원하는 요소의 갯수만큼 index를 만들어야 한다!

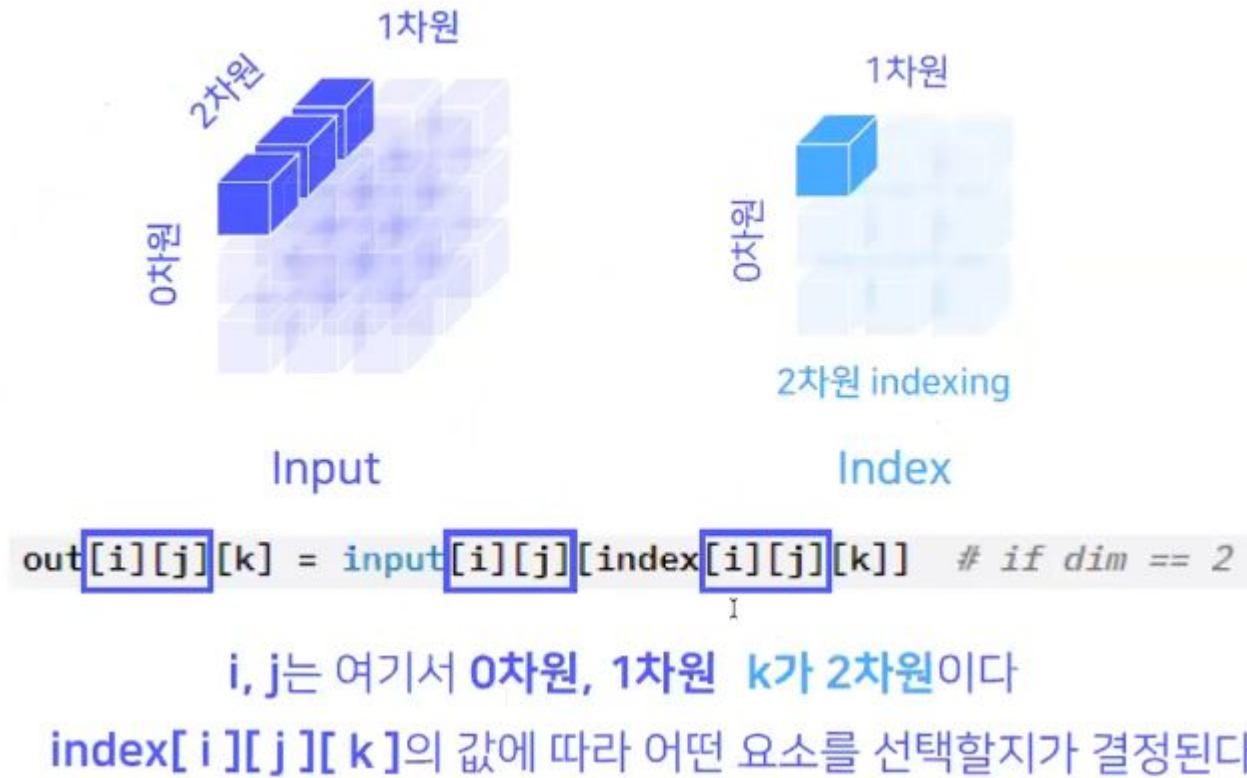


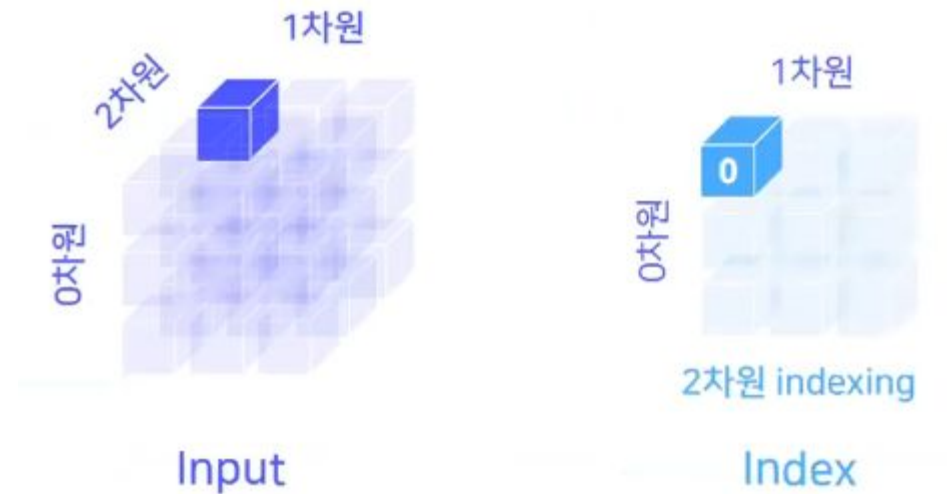
대각 요소를 가져올 경우 1차원 혹은 2차원을 통해 인덱싱이 가능하다!

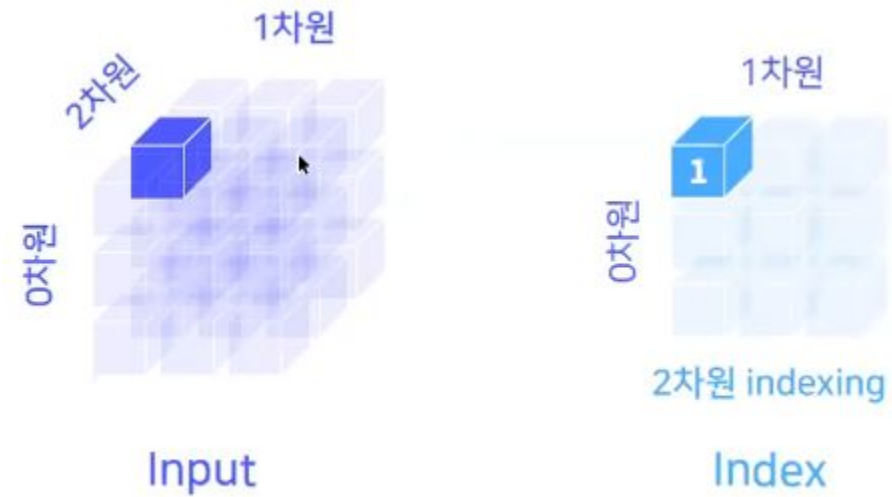
여기서는 시각화 편의를 위해 2차원 indexing을 함!

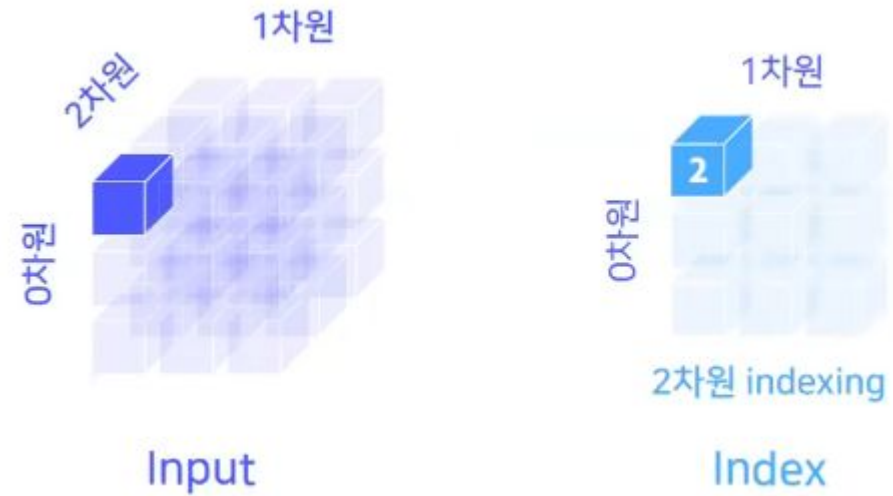


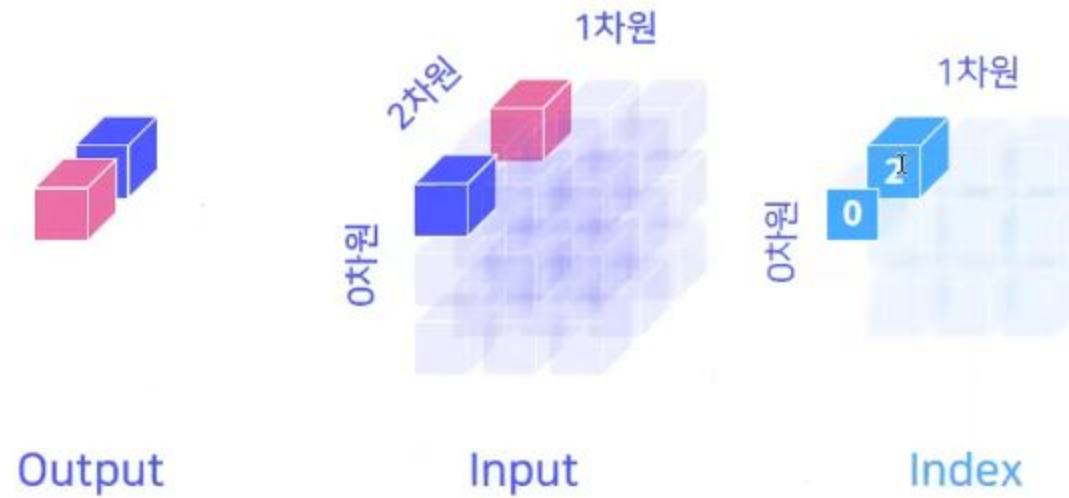
Gather(Input, 2, Index)





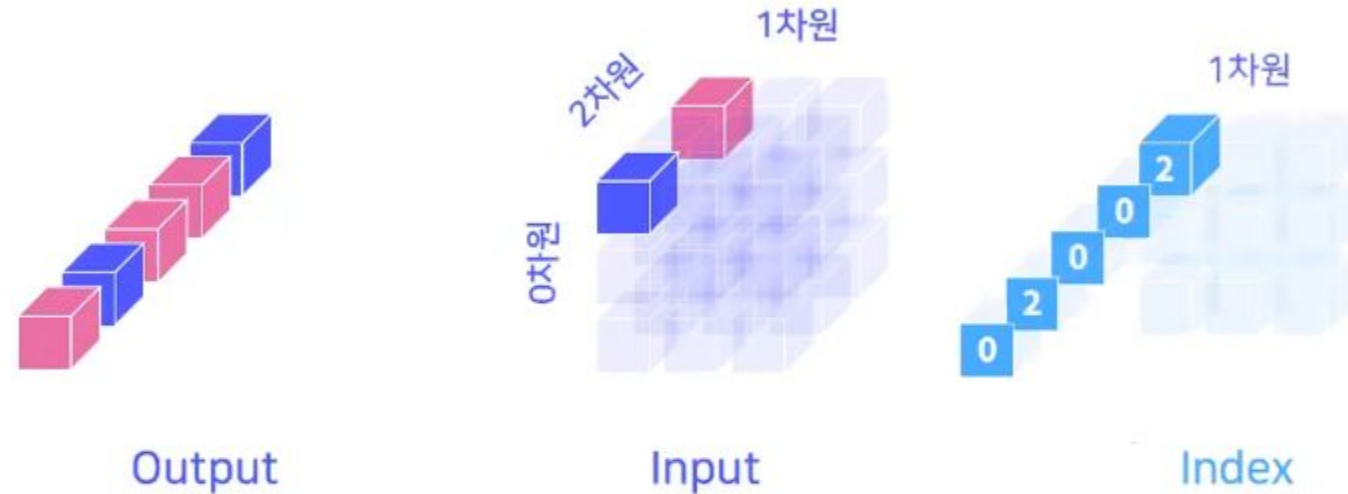




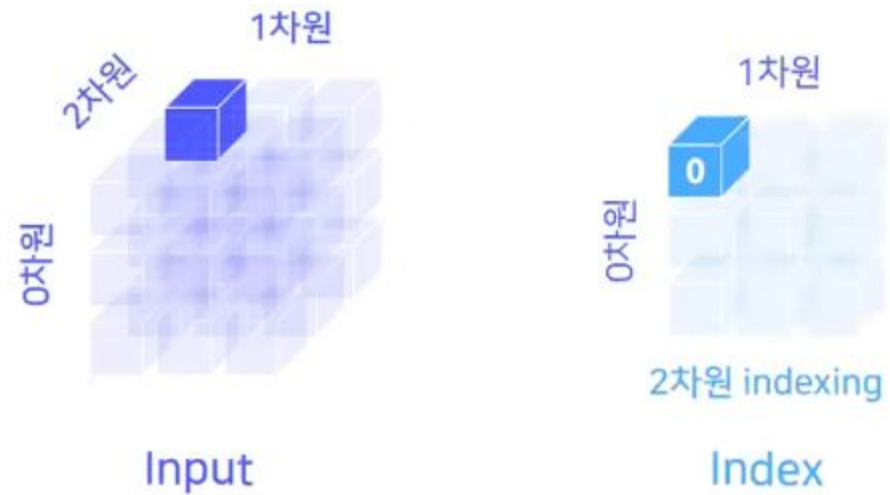


```
out[i][j][k] = input[i][j][index[i][j][k]] # if dim == 2
```

k차원의 값이 2개면 input의 2차원에서 2개의 값을 가져온다!



k값의 갯수만큼 2차원의 값들 중에서 여러가지 값들을 가져올 수 있다
Sampling(?)할 때도 사용할 수 있어보인다

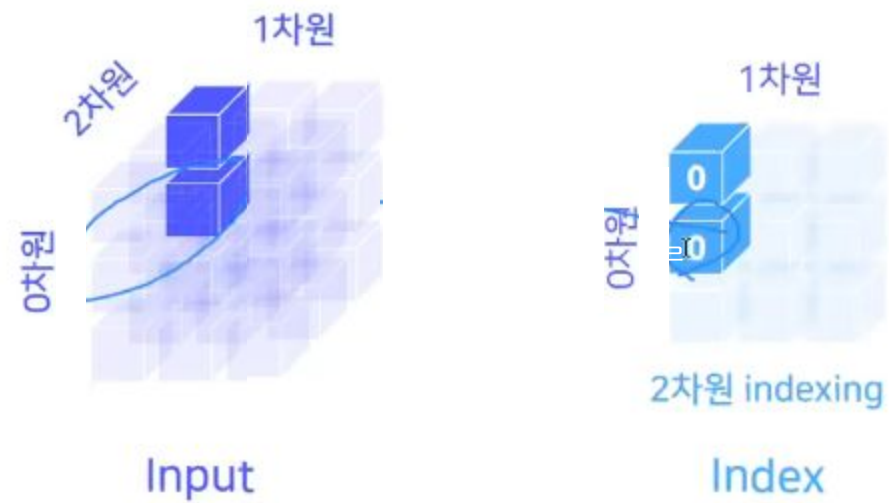


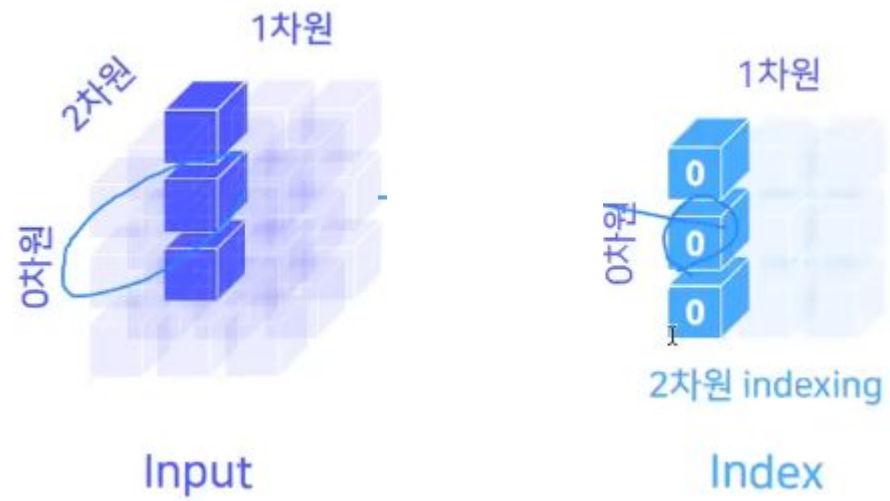


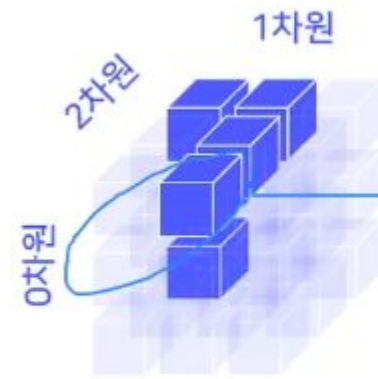
Input



Index





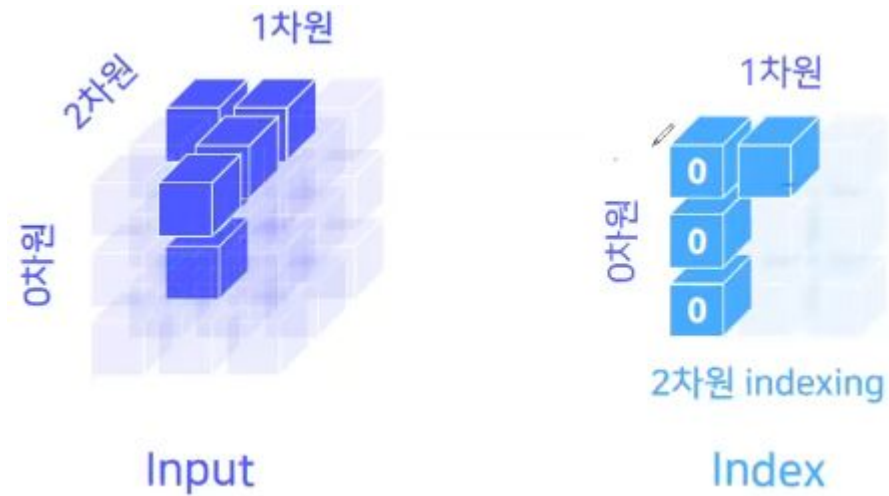


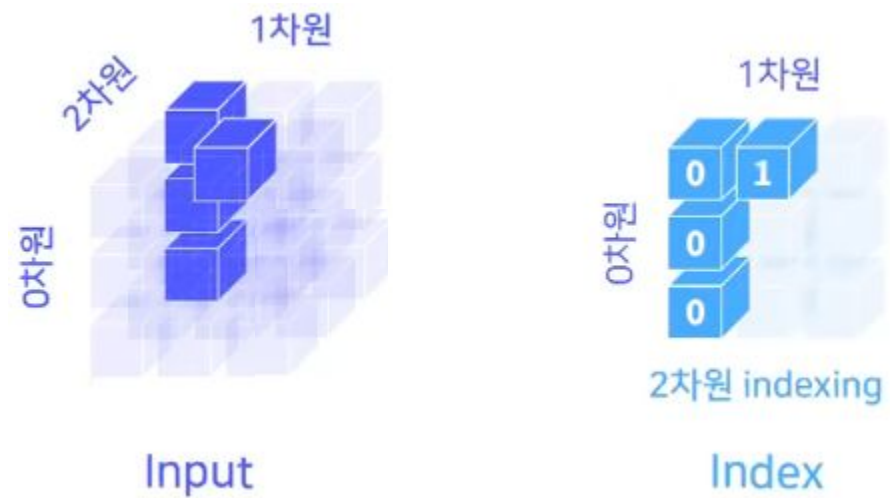
Input

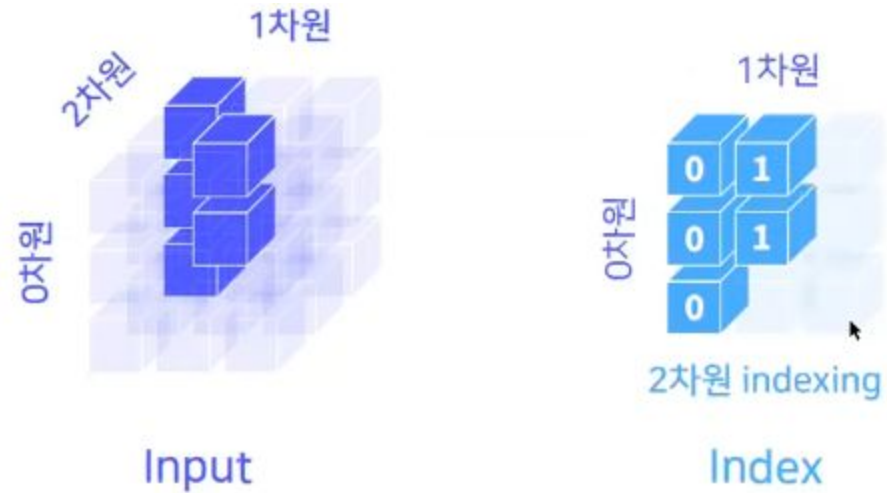


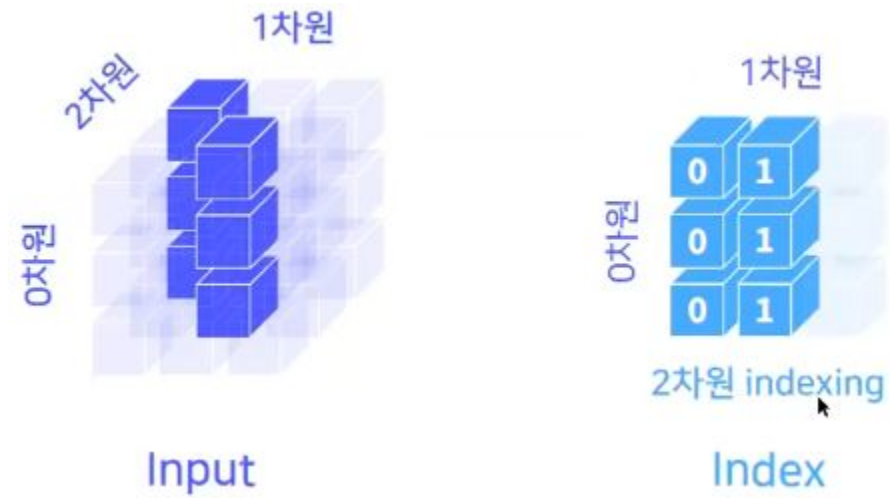
2차원 indexing

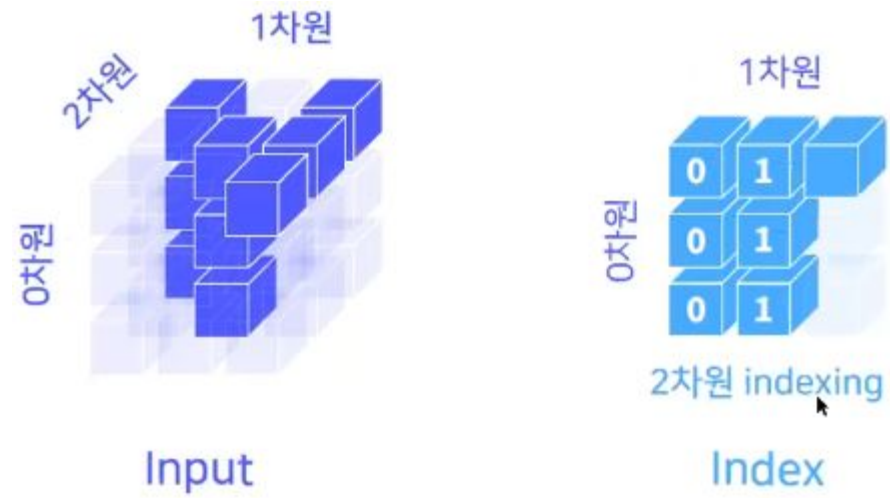
Index

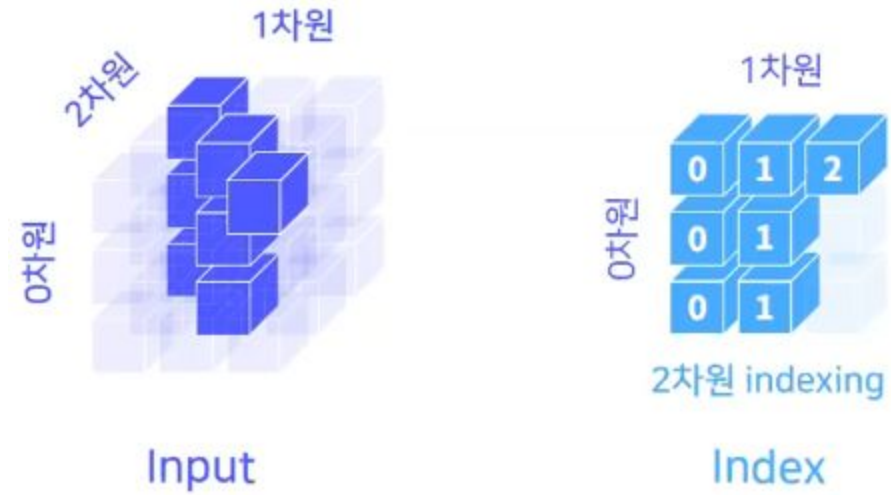


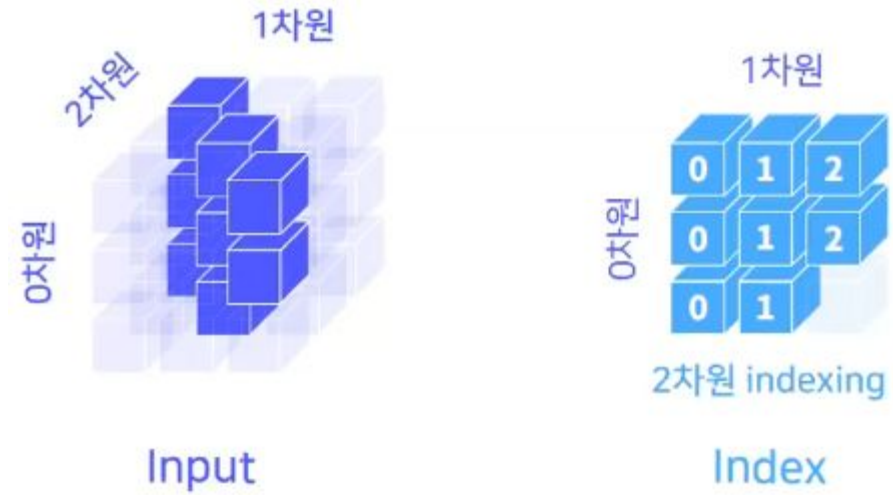


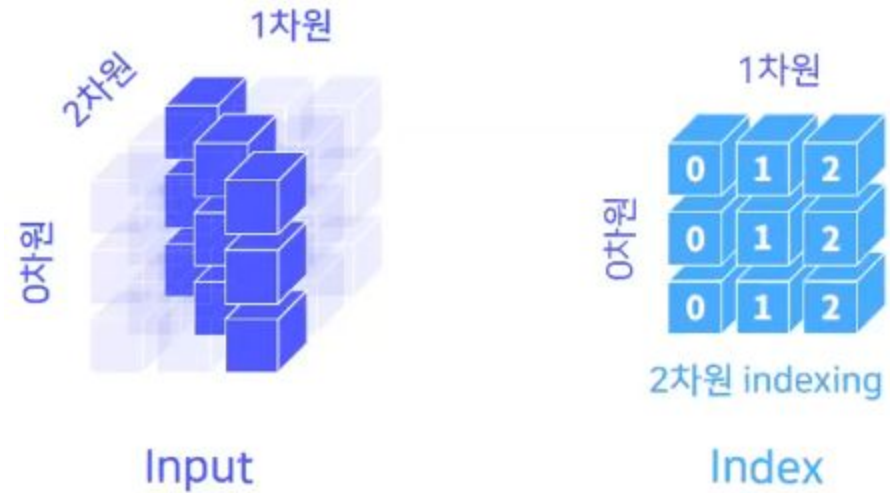


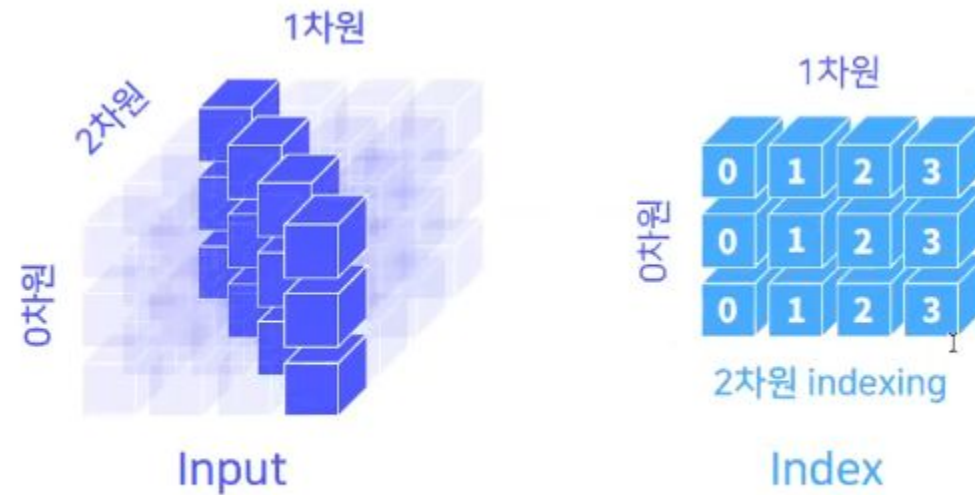




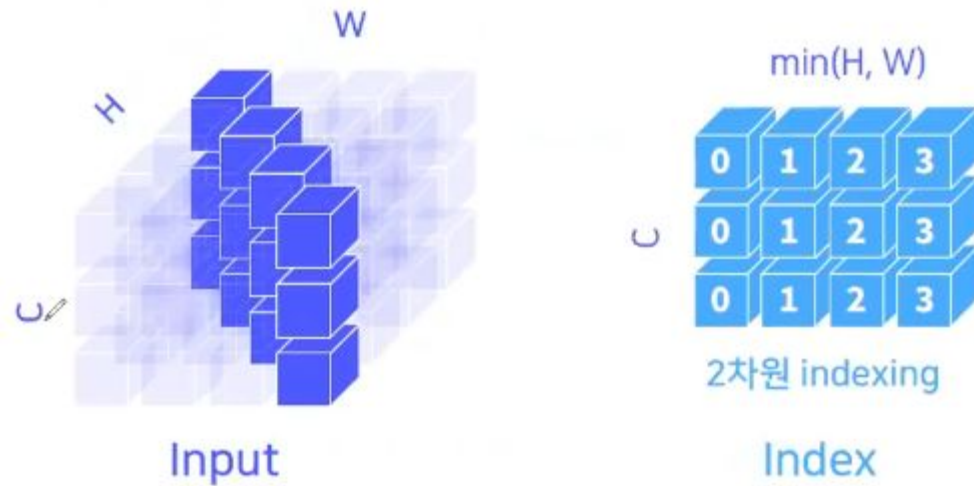


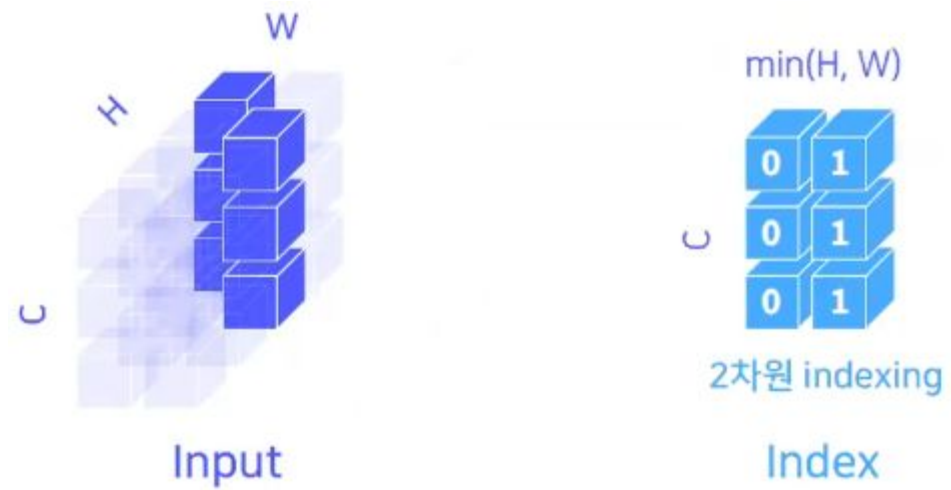


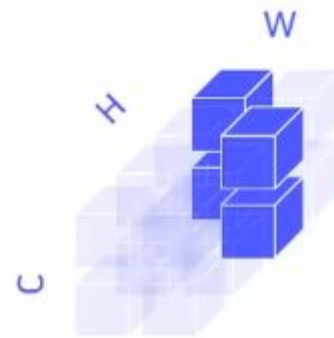




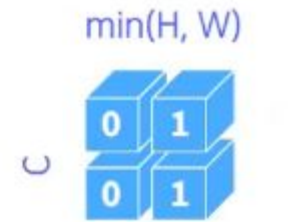
Tensor의 크기가 어떻든 위치를 index를 만들어주면 된다!







Input

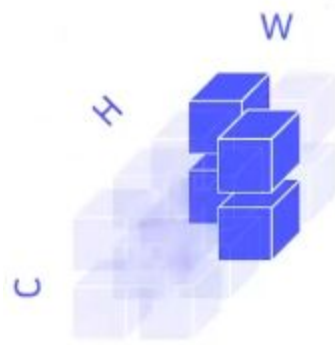


2차원 indexing

Index

```
C, H, W = A.size()
diag_size = min(H, W)

gather_index = torch.arange(diag_size)
```

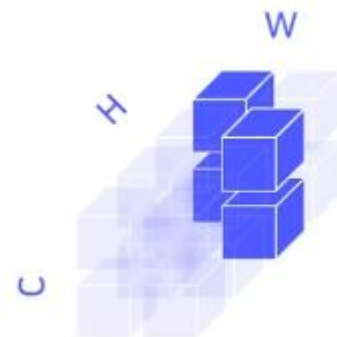


Input

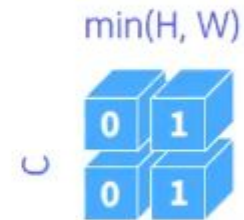


torch.arange

```
.view(diag_size, -1).expand(C, diag_size, 1)
```



Input



view와 expand

3D Gather

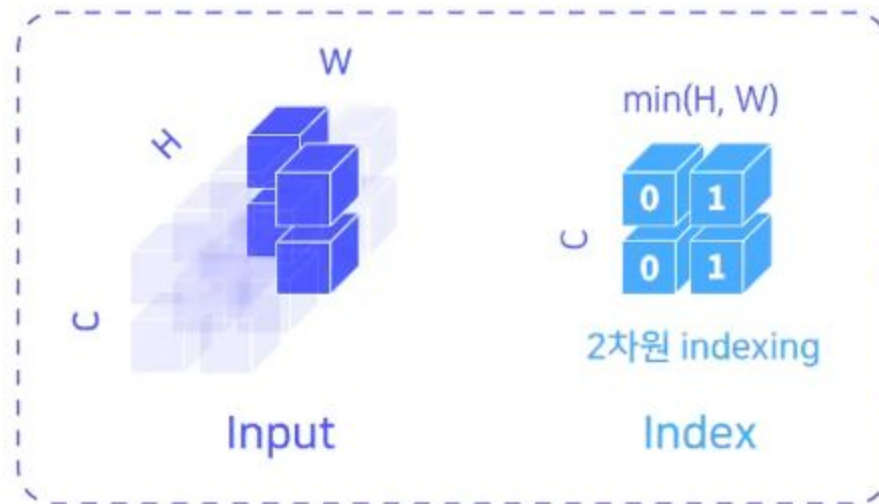
Custom 모델 제작

```
C, H, W = A.size()
diag_size = min(H, W)

gather_index = torch.arange(diag_size).view(diag_size, -1).expand(C, diag_size, 1)
output = torch.gather(A, 2, gather_index)
```



Output



End of Document
Thank You.