



CENG 4480

Embedded System Development & Applications

Lec 04: Direct Conv

Bei Yu

CSE Department, CUHK

byu@cse.cuhk.edu.hk

(Latest update: September 29, 2025)

2025 Fall

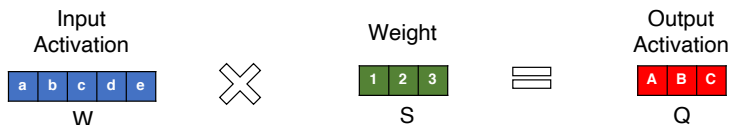


- ① Loop Reordering
- ② Direct Convolution
- ③ Dataflow Optimization



Loop Reordering

1D Convolution Example



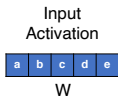
```
for(q=0; q<Q; q++){  
  for (s=0; s<S; s++){  
    OA[q] += IA[q+s] * W[s];  
  }  
}
```

**Output Stationary (OS)
Dataflow**

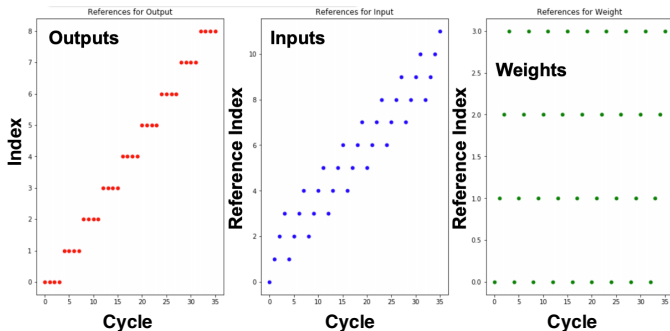
```
for (s=0; s<S; s++){  
  for(q=0; q<Q; q++){  
    OA[q] += IA[q+s] * W[s];  
  }  
}
```

**Weight Stationary (WS)
Dataflow**

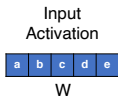
Buffer Access Pattern 1: Output Stationary



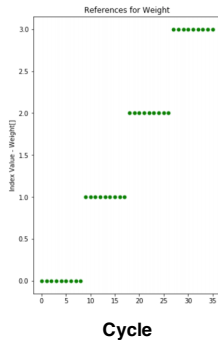
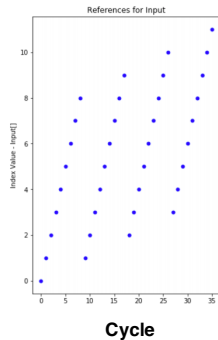
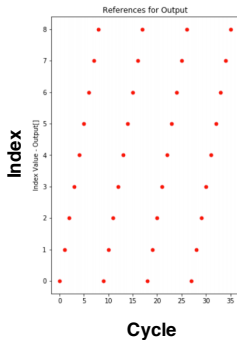
```
for (q=0; q<Q; q++){ // Q =9
    for (s=0; s<S; s++){ // S=4
        OA[q] += IA[q+s] * W[s];
    }
}
```



Buffer Access Pattern 2: Weight Stationary



```
for (s=0; s<S; s++){ // S=4
  for (q=0; q<Q; q++){ // Q =9
    OA[q] += IA[q+s] * W[s];
  }
}
```

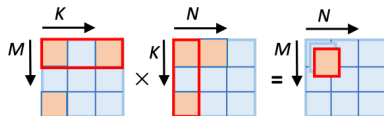


2D Convolution Example



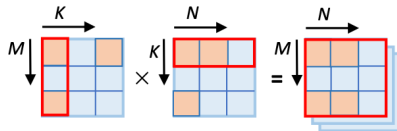
Inner-product
dataflow

```
for m in 0..M
  for n in 0..N
    for k in 0..K
      C[m,n] += A[m,k] * B[k,n]
```



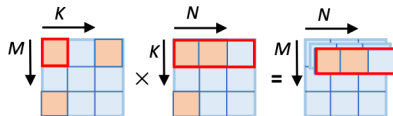
Outer-product
dataflow

```
for k in 0..K
  for m in 0..M
    for n in 0..N
      C[m,n] += A[m,k] * B[k,n]
```



Row-based
dataflow

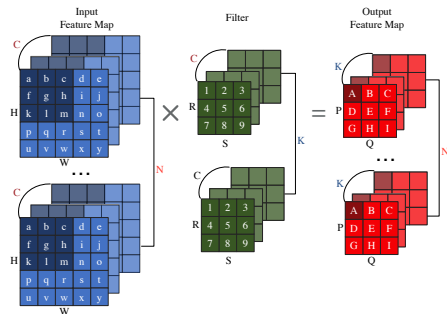
```
for m in 0..M
  for k in 0..K
    for n in 0..N
      C[m,n] += A[m,k] * B[k,n]
```



	InP	OutP	ROW
Input reuse (B)	Poor	Excellent	Poor
Output reuse (C)	Excellent	Poor	Good
Index intersection	Inefficient	Efficient	Efficient
Psum granularity	Scalar	Matrix	Vector



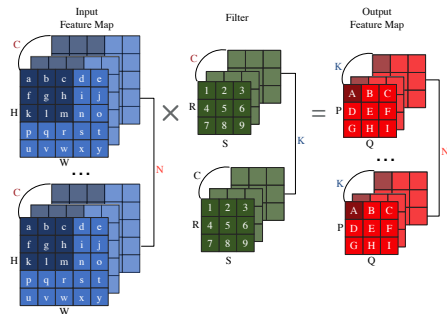
Direct Convolution



```

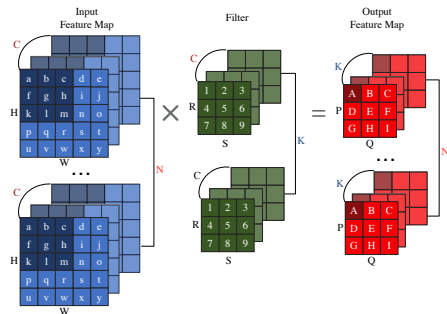
1  for (n=0; n<N; n++) {
2  for (k=0; k<K; k++) {
3  for (p=0; p<P; p++) {
4  for (q=0; q<Q; q++) {
5      OA[n][k][p][q] = 0;
6      for (r=0; r<R; r++) {
7      for (s=0; s<S; s++) {
8      for (c=0; c<C; c++) {
9          h = p * stride - pad + r;
10         w = q * stride - pad + s;
11         OA[n][k][p][q] += IA[n][c][h][w] * W[k][c][r][s];
12     } } } } } } } }
    
```

Direct Convolution: Loop Ordering



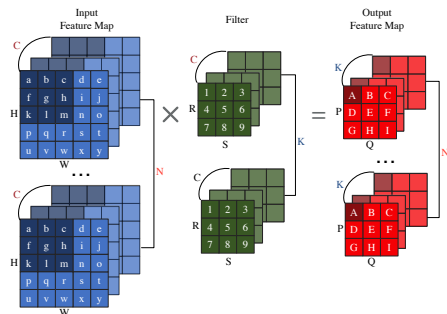
```
1  for (n=0; n<N; n++) {
2  for (r=0; r<R; r++) {
3  for (s=0; s<S; s++) {
4  for (c=0; c<C; c++) {
5  for (k=0; k<K; k++) {
6      float curr_w = W[r][s][c][k];
7      for (p=0; p<P; p++) {
8      for (q=0; q<Q; q++) {
9          h = p * stride - pad + r;
10         w = q * stride - pad + s;
11         OA[n][k][p][q] += IA[n][c][h][w] * curr_w;
12     } } } } } }
```

Direct Convolution: Loop Ordering + Unrolling



```
1  for (n=0; n<N; n++) {
2  for (r=0; r<R; r++) {
3  for (s=0; s<S; s++) {
4      spatial_for (c=0; c<C; c++) {
5      spatial_for (k=0; k<K; k++) {
6          float curr_w = W[r][s][c][k];
7          for (p=0; p<P; p++) {
8              for (q=0; q<Q; q++) {
9                  h = p * stride - pad + r;
10                 w = q * stride - pad + s;
11                 OA[n][k][p][q] += IA[n][c][h][w] * curr_w;
12             } } } } } }
```

Direct Convolution: Loop Ordering + Unrolling + Tiling



```
1  for (n=0; n<N; n++) {
2  for (r=0; r<R; r++) {
3  for (s=0; s<S; s++) {
4  for (c_t=0; c_t<C/16; c_t++) {
5  for (k_t=0; k_t<K/64; k_t++) {
6  spatial_for (c_s=0; c_s<16; c_s++) {
7  spatial_for (k_s=0; k_s<64; k_s++) {
8      int curr_c = c_t * 16 + c_s;
9      int curr_k = k_t * 64 + k_s;
10     float curr_w = W[r][s][curr_c][curr_k];
11     for (p=0; p<P; p++) for (q=0; q<Q; q++) {
12         h = p * stride - pad + r; w = q * stride - pad + s;
13         OA[n][curr_k][p][q] += IA[n][curr_c][h][w] * curr_w;
14     } } } } }
```



Dataflow Optimization

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix} \times \begin{bmatrix} b_{11} & b_{12} & b_{13} & b_{14} \\ b_{21} & b_{22} & b_{23} & b_{24} \\ b_{31} & b_{32} & b_{33} & b_{34} \\ b_{41} & b_{42} & b_{43} & b_{44} \end{bmatrix} = \begin{bmatrix} c_{11} & c_{12} & c_{13} & c_{14} \\ c_{21} & c_{22} & c_{23} & c_{24} \\ c_{31} & c_{32} & c_{33} & c_{34} \\ c_{41} & c_{42} & c_{43} & c_{44} \end{bmatrix}$$

$$\begin{bmatrix} 0 & 0 & 0 & b_{14} \\ 0 & 0 & b_{13} & b_{24} \\ 0 & b_{12} & b_{23} & b_{34} \\ b_{11} & b_{22} & b_{33} & b_{44} \\ b_{21} & b_{32} & b_{43} & 0 \\ b_{31} & b_{42} & 0 & 0 \\ b_{41} & 0 & 0 & 0 \end{bmatrix}$$

