

Windows(1D)

Lecture.5 Conv Layers

- Window Extraction

Windows(1D)

$$\vec{x} = (x_0 \ x_1 \ x_2 \ x_3 \ x_4 \ x_5)$$

$$\mathcal{W}_0 = X[0 : 3] = (x_0 \ x_1 \ x_2) \quad \mathcal{W}_1 = X[1 : 4] = (x_1 \ x_2 \ x_3)$$

$$\mathcal{W}_2 = X[2 : 5] = (x_2 \ x_3 \ x_4) \quad \mathcal{W}_3 = X[3 : 6] = (x_3 \ x_4 \ x_5)$$

$$\mathcal{W}_i \neq (\vec{x}_i \ \vec{x}_{i+1} \ \vec{x}_{i+2})$$

- window은 일정한 간격만큼 앞으로 이동을 하면서 실제로 윈도우 안에 들어있는 값들이 차례대로 변하는걸 윈도우라고 한다.

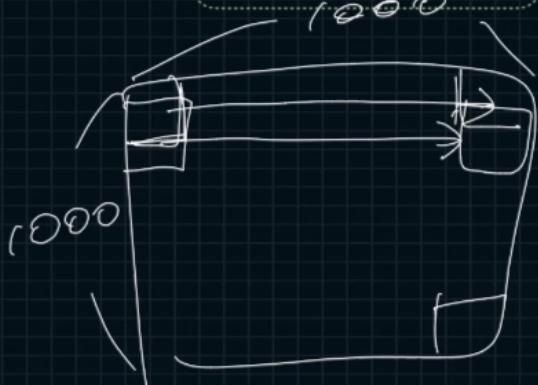
Windows(2D)

Lecture.5 Conv Layers

- Window Extraction

Windows(2D)

$$X = \begin{pmatrix} X[0, 0] & X[0, 1] & X[0, 2] & X[0, 3] \\ X[1, 0] & X[1, 1] & X[1, 2] & X[1, 3] \\ X[2, 0] & X[2, 1] & X[2, 2] & X[2, 3] \\ X[3, 0] & X[3, 1] & X[3, 2] & X[3, 3] \end{pmatrix}$$



$$\mathcal{W}_{0,0} = X[0 : 3, 0 : 3] = \begin{pmatrix} X[0, 0] & X[0, 1] & X[0, 2] \\ X[1, 0] & X[1, 1] & X[1, 2] \\ X[2, 0] & X[2, 1] & X[2, 2] \end{pmatrix}$$

$$\mathcal{W}_{0,1} = X[0 : 3, 1 : 4] = \begin{pmatrix} X[0, 1] & X[0, 2] & X[0, 3] \\ X[1, 1] & X[1, 2] & X[1, 3] \\ X[2, 1] & X[2, 2] & X[2, 3] \end{pmatrix}$$

$$\mathcal{W}_{1,0} = X[1 : 4, 0 : 3] = \begin{pmatrix} X[1, 0] & X[1, 1] & X[1, 2] \\ X[2, 0] & X[2, 1] & X[2, 2] \\ X[3, 0] & X[3, 1] & X[3, 2] \end{pmatrix}$$

$$\mathcal{W}_{1,1} = X[1 : 4, 1 : 4] = \begin{pmatrix} X[1, 1] & X[1, 2] & X[1, 3] \\ X[2, 1] & X[2, 2] & X[2, 3] \\ X[3, 1] & X[3, 2] & X[3, 3] \end{pmatrix}$$

Lecture.5 Conv Layers

- Window Extraction

Windows(2D)

$$X = \begin{pmatrix} X[0, 0] & X[0, 1] & X[0, 2] & X[0, 3] \\ X[1, 0] & \boxed{X[1, 1]} & X[1, 2] & X[1, 3] \\ X[2, 0] & X[2, 1] & X[2, 2] & X[2, 3] \\ X[3, 0] & X[3, 1] & X[3, 2] & X[3, 3] \end{pmatrix}$$

$$\mathcal{W}_{0,0} = X[0 : 3, 0 : 3] = \begin{pmatrix} X[0, 0] & X[0, 1] & X[0, 2] \\ X[1, 0] & X[1, 1] & X[1, 2] \\ X[2, 0] & X[2, 1] & X[2, 2] \end{pmatrix}$$

$$\mathcal{W}_{0,1} = X[0 : 3, 1 : 4] = \begin{pmatrix} X[0, 1] & X[0, 2] & X[0, 3] \\ X[1, 1] & X[1, 2] & X[1, 3] \\ X[2, 1] & X[2, 2] & X[2, 3] \end{pmatrix}$$

$$\mathcal{W}_{i,j} = \begin{pmatrix} X[i, j] & X[i, j+1] & X[i, j+2] \\ X[i+1, j] & X[i+1, j+1] & X[i+1, j+2] \\ X[i+2, j] & X[i+2, j+1] & X[i+2, j+2] \end{pmatrix}$$

$$\mathcal{W}_{1,0} = X[1 : 4, 0 : 3] = \begin{pmatrix} X[1, 0] & X[1, 1] & X[1, 2] \\ X[2, 0] & X[2, 1] & X[2, 2] \\ X[3, 0] & X[3, 1] & X[3, 2] \end{pmatrix}$$

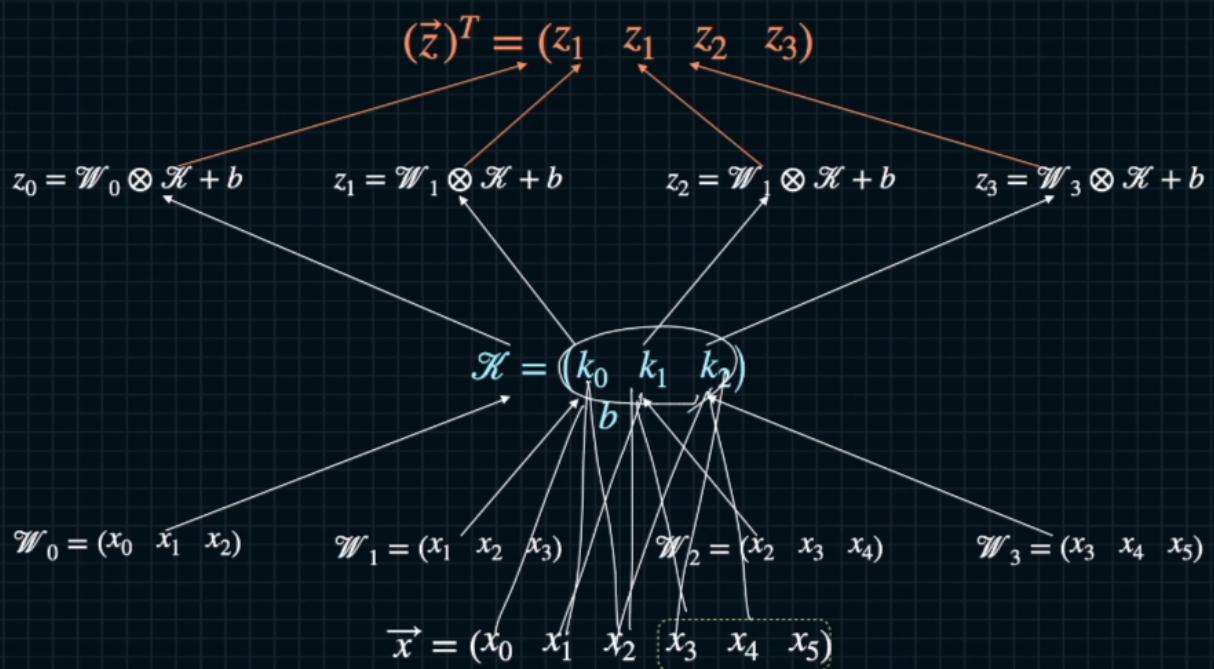
$$\mathcal{W}_{1,1} = X[1 : 4, 1 : 4] = \begin{pmatrix} X[1, 1] & X[1, 2] & X[1, 3] \\ X[2, 1] & X[2, 2] & X[2, 3] \\ X[3, 1] & X[3, 2] & X[3, 3] \end{pmatrix}$$

2차원인 행렬도 1차원에서 했던거랑 비슷하다. Window가 row에서 3 column에서 3만큼씩 앞으로 이동하면서 연산을 한다. 초록색으로 된게 window를 일반화 한것이다.

Window Formularization

Lecture.5 Conv Layers

- Computations of Conv Layer



Lecture.5 Conv Layers

- Window Extraction

Window Formularization

$$X \in \mathbb{R}^{n_H \times n_W}, \mathcal{W}_{i,j} \in \mathbb{R}^{f \times f}$$

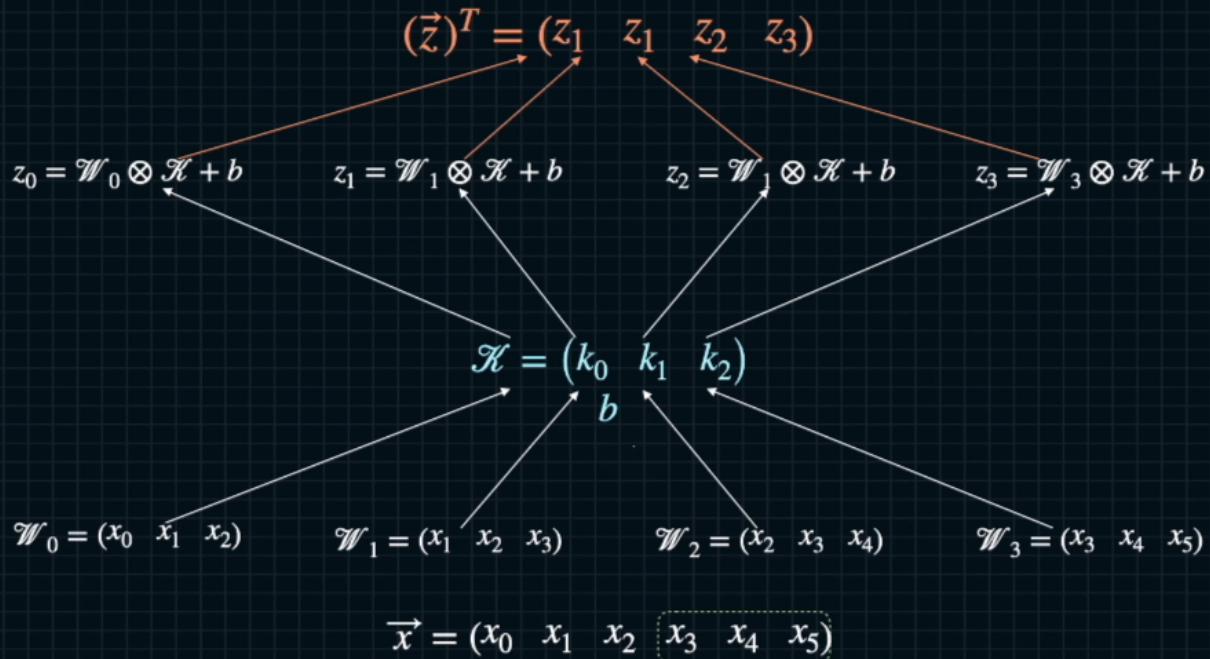
$$X = \begin{pmatrix} X[0, 0] & X[0, 1] & \dots & X[0, n_W - f] & \dots & X[0, n_W - 1] \\ X[1, 0] & X[1, 1] & \dots & X[1, n_W - f] & \dots & X[1, n_W - 1] \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ X[n_H - f, 0] & X[n_H - f, 1] & \dots & X[n_H - f, n_W - f] & \dots & X[n_H - f, n_W - 1] \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ X[n_H - 1, 0] & X[n_H - 1, 1] & \dots & X[n_H - 1, n_W - f] & \dots & X[n_H - 1, n_W - 1] \end{pmatrix}$$

$$\mathcal{W}_{i,j} X[i : i + f, j : j + f] = \begin{pmatrix} X[i, j] & X[i, j + 1] & \dots & X[i, j + (f - 1)] \\ X[i + 1, j] & X[i + 1, j + 1] & \dots & X[i + 1, j + (f - 1)] \\ \vdots & \vdots & \ddots & \vdots \\ X[i + (f - 1), j] & X[i + (f - 1), j + 1] & \dots & X[i + (f - 1), j + (f - 1)] \end{pmatrix}$$

$$0 \leq i \leq n_H - f, \quad 0 \leq j \leq n_W - f$$

Lecture.5 Conv Layers

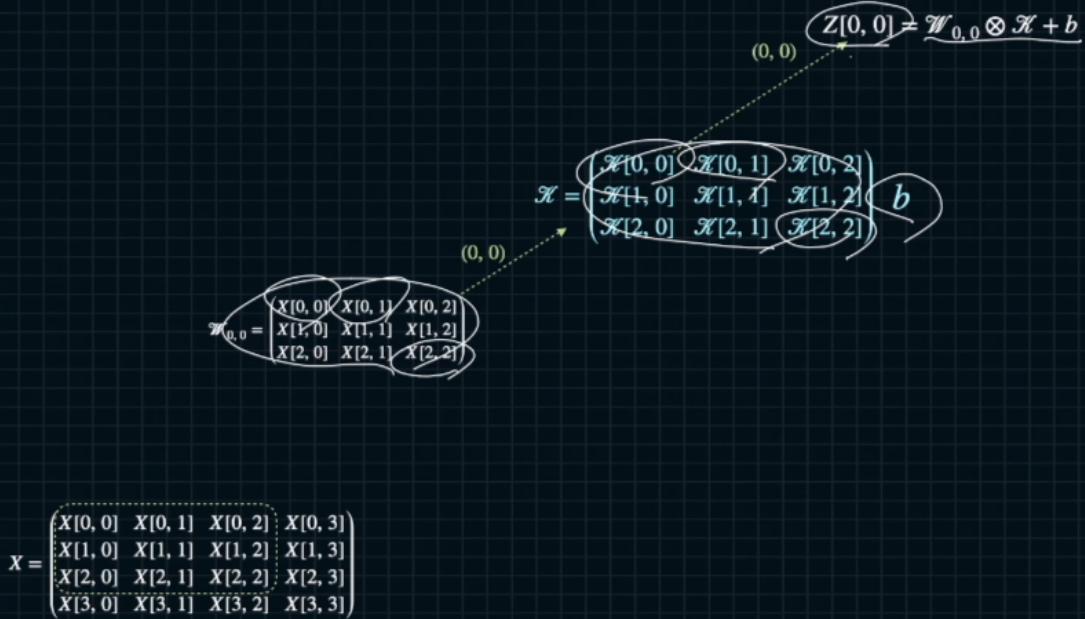
- Computations of Conv Layer



- Dense layer랑 conv layer랑 굳이 비교하자면, 연산이 되는 대상이 정해져있다. 내가 filter 가 바라보고 있는 window 부분이 앞으로 이동하면서 연산이 되기 때문에 연산이 정해져 있다고 생각하면 된다. window의 부분을 receptive field라고 불린다.

Lecture.5 Conv Layers

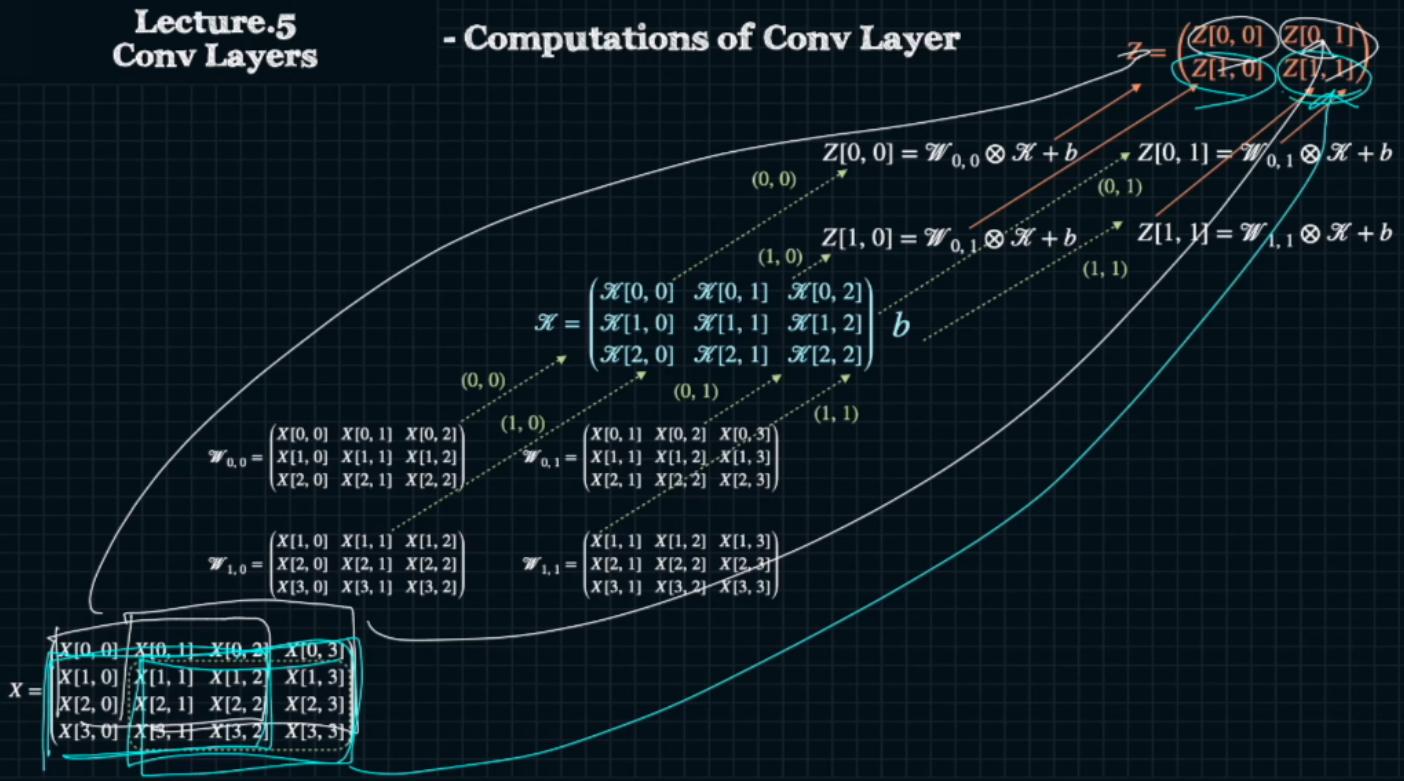
- Computations of Conv Layer



- 3 x 3 filter와 window에서 뽑힌 값들이랑 행렬 연산과 bias를 더한 Affine transformation을 하게되면 행렬의 첫 번째 값으로 가지게 된다 두 번째도 마찬가지 마지막까지

Lecture.5 Conv Layers

- Computations of Conv Layer



- 0이미지의 차원은 변하지 않고 Height Width은 변하는걸 볼 수 있다.

Lecture.5 Conv Layers

- Computations of Conv Layer

$$X \in \mathbb{R}^{n_H \times n_W}$$

$$\mathcal{W}_{i,j} \in \mathbb{R}^{f \times f}$$

$$\mathcal{K} \in \mathbb{R}^{f \times f}, b \in \mathbb{R}$$

$$Z[i, j] = \mathcal{W}_{i,j} \otimes \mathcal{K} + b$$

$$Z = \text{Conv2D}(X; \mathcal{K}, b) = \begin{pmatrix} \mathcal{W}_{0,0} \otimes \mathcal{K} + b & \mathcal{W}_{0,1} \otimes \mathcal{K} + b & \dots & \mathcal{W}_{0,n_W-f} \otimes \mathcal{K} + b \\ \mathcal{W}_{1,0} \otimes \mathcal{K} + b & \mathcal{W}_{1,1} \otimes \mathcal{K} + b & \dots & \mathcal{W}_{1,n_W-f} \otimes \mathcal{K} + b \\ \vdots & \vdots & \ddots & \vdots \\ \mathcal{W}_{n_H-f,0} \otimes \mathcal{K} + b & \mathcal{W}_{n_H-f,1} \otimes \mathcal{K} + b & \dots & \mathcal{W}_{n_H-f,n_W-f} \otimes \mathcal{K} + b \end{pmatrix}$$

$$n'_H = \underbrace{n_H}_{\textcircled{n}_H} - \boxed{f} + 1$$

$$n'_W = \underbrace{n_W}_{\textcircled{n}_W} - \boxed{f} + 1$$

- 초록색 공식은 외울것

Lecture.5 Conv Layers

- Computations of Conv Layer

$$Z[i, j] = \mathcal{W}_{i,j} \otimes \mathcal{K} + b$$

$$A[i, j] = \sigma(Z[i, j])$$

$$A = \text{Conv2D}(X; \mathcal{K}, b; \sigma) = \begin{pmatrix} \sigma(\mathcal{W}_{0,0} \otimes \mathcal{K} + b) & \sigma(\mathcal{W}_{0,1} \otimes \mathcal{K} + b) & \dots & \sigma(\mathcal{W}_{0,n_W-f} \otimes \mathcal{K} + b) \\ \sigma(\mathcal{W}_{1,0} \otimes \mathcal{K} + b) & \sigma(\mathcal{W}_{1,1} \otimes \mathcal{K} + b) & \dots & \sigma(\mathcal{W}_{1,n_W-f} \otimes \mathcal{K} + b) \\ \vdots & \vdots & \ddots & \vdots \\ \sigma(\mathcal{W}_{n_H-f,0} \otimes \mathcal{K} + b) & \sigma(\mathcal{W}_{n_H-f,1} \otimes \mathcal{K} + b) & \dots & \sigma(\mathcal{W}_{n_H-f,n_W-f} \otimes \mathcal{K} + b) \end{pmatrix}$$

- Filter와 window가 Colerration 연산을 하고 bias를 더하면 Afiine transformation이 되고 이것을 z 값이 된다. 그래서 이 z값에다가 Activation function을 통과해야지 conv layer가 된다. 행렬의 각 원소에 activation function을 통과한 모습이다.