Matrix Operations I

- For efficient implementations, we should conduct convolutional operations by matrix-matrix and matrix-vector operations.
- Let’s collect images of all channels as the input

\[
Z^{in,i} = \begin{bmatrix}
    z_{1,1,1}^i & z_{2,1,1}^i & \cdots & z_{a_{in},b_{in},1}^i \\
    \vdots & \vdots & \ddots & \vdots \\
    z_{1,1,d_{in}}^i & z_{2,1,d_{in}}^i & \cdots & z_{a_{in},b_{in},d_{in}}^i
\end{bmatrix} \in \mathbb{R}^{d_{in} \times a_{in}b_{in}}.
\]
Let all filters

\[ W = \begin{bmatrix}
  w_{1,1,1} & w_{1,2,1} & \cdots & w_{1,h,h,d_{in}} \\
  \vdots & \vdots & \ddots & \vdots \\
  w_{d_{out},1,1} & w_{d_{out},2,1} & \cdots & w_{d_{out},h,h,d_{in}}
\end{bmatrix} \in \mathbb{R}^{d_{out} \times hh_{d_{in}}}
\]

be variables (parameters) of the current layer.
Matrix Operations III

- Usually a bias term is considered

\[ b = \begin{bmatrix} b_1 \\ \vdots \\ b_{\text{out}} \end{bmatrix} \in \mathbb{R}^{d_{\text{out}} \times 1} \]

- Operations at a layer

\[ S_{\text{out},i} = W \phi(Z_{\text{in},i}) + b_1^{T} a_{\text{out}} b_{\text{out}} \]
\[ \in \mathbb{R}^{d_{\text{out}} \times a_{\text{out}} b_{\text{out}}} \]  

(1)
where

\[
\mathbf{1}_{a_{\text{out}} \times b_{\text{out}}} = \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix} \in \mathbb{R}^{a_{\text{out}} \times b_{\text{out}} \times 1}.
\]

\(\phi(Z^{\text{in},i})\) collects all sub-images in \(Z^{\text{in},i}\) into a matrix.
Specifically,

\[ \phi(Z^{in,i}) = \begin{bmatrix}
    z^i_{1,1,1} & z^i_{1+s,1,1} \\
    z^i_{2,1,1} & z^i_{2+s,1,1} \\
    \vdots & \vdots & \ddots \\
    z^i_{h,h,1} & z^i_{h+s,h,1} \\
    \vdots & \vdots & \\
    z^i_{h,h,d^{in}} & z^i_{h+s,h,d^{in}}
\end{bmatrix} \in \mathbb{R}^{hhd^{in}} \times a^{out} b^{out} \]
Next, an activation function scales each element of $S^{\text{out},i}$ to obtain the output matrix $Z^{\text{out},i}$.

$$Z^{\text{out},i} = \sigma(S^{\text{out},i}) \in \mathbb{R}^{d_{\text{out}} \times a_{\text{out}} b_{\text{out}}}. \quad (2)$$

For CNN, commonly the following RELU activation function

$$\sigma(x) = \max(x, 0) \quad (3)$$

is used.

Later we need that $\sigma(x)$ is differentiable, but the RELU function is not.
Past works such as Krizhevsky et al. (2012) assume

$$\sigma'(x) = \begin{cases} 
1 & \text{if } x > 0 \\
0 & \text{otherwise}
\end{cases}$$
The Function $\phi(Z^{in,i})$ I

- In the matrix-matrix product

$$W\phi(Z^{in,i}),$$

each element is the inner product between a filter and a sub-image

- Clearly $\phi$ is a linear mapping, so there exists a 0/1 matrix $P_\phi$ such that

$$\phi(Z^{in,i}) \equiv \text{mat}(P_\phi \text{vec}(Z^{in,i}))_{hhd^{in} \times a^{out}b^{out}}, \forall i, \quad (4)$$
The Function $\phi(Z^{in,i}) II$

- $\text{vec}(M)$: all $M$'s columns concatenated to a vector $\mathbf{v}$

$$\text{vec}(M) = \begin{bmatrix} M_{:,1} \\ \vdots \\ M_{:,b} \end{bmatrix} \in \mathbb{R}^{ab \times 1}, \text{ where } M \in \mathbb{R}^{a \times b}$$

- $\text{mat}(\mathbf{v})$ is the inverse of $\text{vec}(M)$

$$\text{mat}(\mathbf{v})_{a \times b} = \begin{bmatrix} v_1 & v_{(b-1)a+1} \\ \vdots & \vdots \\ v_a & v_{ba} \end{bmatrix} \in \mathbb{R}^{a \times b}, \quad (5)$$
The Function $\phi(z^{in}, i)$ III

where

$$v \in \mathbb{R}^{ab \times 1}.$$  

- $P_\phi$ is a huge matrix:

$$P_\phi \in \mathbb{R}^{hhd^{in} \times a^{out} b^{out} \times d^{in} a^{in} b^{in}}$$

and

$$\phi : \mathbb{R}^{d^{in} \times a^{in} b^{in}} \rightarrow \mathbb{R}^{hhd^{in} \times a^{out} b^{out}}$$

- The representation of using $P_\phi$ makes some subsequent derivations easier
The Function $\phi(Z^{in}, i)$

- Past works using the form (4) include, for example, Vedaldi and Lenc (2015)
We collect all weights to a vector variable $\theta$.

$$\theta = \begin{bmatrix}
\text{vec}(W^1) \\
b^1 \\
\vdots \\
\text{vec}(W^L) \\
b^L
\end{bmatrix} \in \mathbb{R}^n, \quad n : \text{total \# variables}$$

The output of the last layer $L$ is a vector $z^{L+1,i}(\theta)$.

Consider any loss function such as the squared loss

$$\xi_i(\theta) = \|z^{L+1,i}(\theta) - y^i\|^2.$$
The optimization problem is

$$\min_{\theta} f(\theta),$$

where

$$f(\theta) = \frac{1}{2C} \theta^T \theta + \frac{1}{l} \sum_{i=1}^{l} \xi(z^{L+1,i}(\theta); y^i, Z^{1,i})$$

$C$: regularization parameter.

The formulation is almost the same as that for fully connected networks.
Note that we divide the sum of training losses by the number of training data. Thus the second term becomes the average training loss.