Summary of a Convolutional Layer I

- Padding and pooling are optional in a convolutional layer, but they are frequently used
- Thus we discuss details of considering all operations together.
- The whole convolutional layer involves the following procedure:

$$
\begin{aligned}
Z^{m,i} &\rightarrow \text{padding} \\
&\rightarrow \text{convolutional operations} \\
&\rightarrow \text{pooling} \ \rightarrow Z^{m+1,i},
\end{aligned} \tag{1}
$$

Summary of a Convolutional Layer II

where $Z^{m,i}$ and $Z^{m+1,i}$ are input and output of the mth layer, respectively.

- Let the following symbols denote image sizes at different stages of the convolutional layer.

$$\begin{aligned} a^m,\ b^m &: \text{ size in the beginning} \\ a^m_{\text{pad}},\ b^m_{\text{pad}} &: \text{ size after padding} \\ a^m_{\text{conv}},\ b^m_{\text{conv}} &: \text{ size after convolution.} \end{aligned}$$

- The following table indicates how these values are $a^{\text{in}}, b^{\text{in}}, d^{\text{in}}$ and $a^{\text{out}}, b^{\text{out}}, d^{\text{out}}$ at different stages.

Summary of a Convolutional Layer III

Operation	Input	Output
Padding	$Z^{m,i}$	$\mathrm{pad}(Z^{m,i})$
Convolution	$\mathrm{pad}(Z^{m,i})$	$S^{m,i}$
Convolution	$S^{m,i}$	$\sigma(S^{m,i})$
Pooling	$\sigma(S^{m,i})$	$Z^{m+1,i}$

Operation	$a^{\text{in}},\ b^{\text{in}},\ d^{\text{in}}$	$a^{\text{out}},\ b^{\text{out}},\ d^{\text{out}}$
Padding	$a^m,\ b^m,\ d^m$	$a^m_{\text{pad}},\ b^m_{\text{pad}},\ d^m$
Convolution	$a^m_{\text{pad}},\ b^m_{\text{pad}},\ d^m$	$a^m_{\text{conv}},\ b^m_{\text{conv}},\ d^{m+1}$
Convolution	$a^m_{\text{conv}},\ b^m_{\text{conv}},\ d^{m+1}$	$a^m_{\text{conv}},\ b^m_{\text{conv}},\ d^{m+1}$
Pooling	$a^m_{\text{conv}},\ b^m_{\text{conv}},\ d^{m+1}$	$a^{m+1},\ b^{m+1},\ d^{m+1}$

Summary of a Convolutional Layer IV

- Let the filter size, mapping matrices and weight matrices at the *m*th layer be

$$h^m,\ P_{\text{pad}}^m,\ P_{\phi}^m,\ P_{\text{pool}}^{m,i},\ W^m,\ \boldsymbol{b}^m.$$

- Then all operations can be summarized as

$$S^{m,i} = W^m \text{mat}(P_{\phi}^m P_{\text{pad}}^m \text{vec}(Z^{m,i}))_{h^m h^m d^m \times a_{\text{conv}}^m b_{\text{conv}}^m} + \boldsymbol{b}^m \mathbb{1}_{a^{\text{conv}} b^{\text{conv}}}^T$$

$$Z^{m+1,i} = \text{mat}(P_{\text{pool}}^{m,i} \text{vec}(\sigma(S^{m,i})))_{d^{m+1} \times a^{m+1} b^{m+1}}, \quad (2)$$

Fully-Connected Layer I

- Assume L^C is the number of convolutional layers
- Input vector of the first fully-connected layer:

$$z^{m,i} = \text{vec}(Z^{m,i}), \ i = 1, \ldots, I, \ m = L^c + 1.$$

- In each of the fully-connected layers ($L^c < m \leq L$), we consider weight matrix and bias vector between layers m and $m + 1$.

Fully-Connected Layer II

- Weight matrix:

$$W^m = \begin{bmatrix} w_{11}^m & w_{12}^m & \cdots & w_{1n_m}^m \\ w_{21}^m & w_{22}^m & \cdots & w_{2n_m}^m \\ \vdots & \vdots & \vdots & \vdots \\ w_{n_{m+1}1}^m & w_{n_{m+1}2}^m & \cdots & w_{n_{m+1}n_m}^m \end{bmatrix}_{n_{m+1} \times n_m} \tag{3}$$

- Bias vector

$$\boldsymbol{b}^m = \begin{bmatrix} b_1^m \\ b_2^m \\ \vdots \\ b_{n_{m+1}}^m \end{bmatrix}_{n_{m+1} \times 1}$$

Fully-Connected Layer III

Here n_m and n_{m+1} are the numbers of nodes in layers m and $m+1$, respectively.

- If $\boldsymbol{z}^{m,i} \in R^{n_m}$ is the input vector, the following operations are applied to generate the output vector $\boldsymbol{z}^{m+1,i} \in R^{n_{m+1}}$.

$$\boldsymbol{s}^{m,i} = W^m \boldsymbol{z}^{m,i} + \boldsymbol{b}^m, \tag{4}$$

$$z_j^{m+1,i} = \sigma(s_j^{m,i}),\ j = 1, \ldots, n_{m+1}. \tag{5}$$

Challenges in NN Optimization

- The objective function is non-convex. It may have many local minima
- It's known that global optimization is much more difficult than local minimization
- The problem structure is very complicated