Lecture Notes on Deep Learning

I. Notation

Let $i \in \mathbb{Z}$, $j \in \mathbb{Z}$, and $i \leq j$. Then

$$\begin{cases} [i:j] \triangleq \{i, i+1, \cdots, j-1, j\} \\ (i:j) \triangleq \{i+1, \cdots, j-1\} \\ (i:j] \triangleq \{i+1, \cdots, j-1, j\} \\ [i:j) \triangleq \{i, i+1, \cdots, j-1\} \end{cases}. \tag{1}$$

The logistic sigmoid function is defined as

$$\phi(z) = \frac{1}{1+e^{-z}}. \tag{2}$$

Obviously,

$$\frac{\partial \phi(z)}{\partial z} = \phi(z)(1-\phi(z)). \tag{3}$$

The rectifier function is defined as

$$\varphi(z) = \max(0, z) \geq 0. \tag{4}$$

Obviously,

$$\frac{\partial \varphi(z)}{\partial z} = \mathbf{1}_{\varphi(z)>0} = \begin{cases} 0, & \varphi(z) = 0 \\ 1, & \varphi(z) > 0 \end{cases}. \tag{5}$$

Let us define $x_1^n \triangleq (x_1, \cdots, x_n)$.

$$\lfloor x \rfloor_a \triangleq \max(a, x). \tag{6}$$

$$\lceil x \rceil^b \triangleq \min(b, x). \tag{7}$$

II. BACKPROPAGATION

Consider an Artificial Neural Network (ANN) including $(h+2)$ layers. The 1-th layer is the input layer; the $(h+2)$-th layer is the output layer; and the intermediate h layers are hidden layers.

The l-th layer, where $l \in [1 : (h+2)]$, includes m_l nodes. Especially, the input layer includes $m_1 = n_x$ input nodes and the output layer includes $m_{h+2} = n_y$ output nodes.

The i-th node of the l-th layer is denoted by $z_{l,i}$, where $l \in [1 : (h+2)]$ and $i \in [1 : m_l]$. Especially, $z_{1,i} = x_i$ is the i-th input node and $z_{h+2,i} = y_i$ is the i-th output node. The l-th layer node vector is denoted by $\mathbf{z}_l \in \mathbb{R}^{m_l}$. Especially, $\mathbf{z}_1 = \mathbf{x} \in \mathbb{R}^{n_x}$ and $\mathbf{z}_{h+2} = \mathbf{y} \in \mathbb{R}^{n_y}$.

Let $w_{l,i,j}$ denote the weight of the connection between the i-th node of the l-th layer and the j-th node of the $(l+1)$-th layer, where $l \in [1 : (h+1)]$, $i \in [1 : m_l]$, and $j \in [1 : m_{l+1}]$. The l-th layer weight matrix is denoted by $\mathbf{W}_l \in \mathbb{R}^{m_l \times m_{l+1}}$.

Let $b_{l,j}$ denote the bias associated with the j-th node of the $(l+1)$-th layer, where $l \in [1 : (h+1)]$ and $j \in [1 : m_{l+1}]$. The l-th bias vector is denoted by $\mathbf{b}_l \in \mathbb{R}^{m_{l+1}}$.

For $l \in [1 : (h+1)]$ and $j \in [1 : m_{l+1}]$, we define

$$\psi_{l+1,j} \triangleq b_{l,j} + \sum_{i=1}^{m_l} w_{l,i,j} z_{l,i}. \tag{8}$$

Then $z_{l+1,j} = \phi(\psi_{l+1,j})$. Alternatively, $\mathbf{z}_{l+1} = \phi(\mathbf{W}_l^\top \mathbf{z}_l + \mathbf{b}_l)$. Especially, we have

$$\begin{cases} \psi_{2,j} = b_{1,j} + \sum_{i=1}^{n_x} w_{1,i,j} x_i \\ y_j = z_{h+2,j} = \phi(\psi_{h+2,j}) \end{cases}. \tag{9}$$

It is easy to get

$$\begin{cases} \frac{\partial z_{l+1,j}}{\partial b_{l,j}} = \frac{\partial \phi(\psi_{l+1,j})}{\partial \psi_{l+1,j}} \cdot \frac{\partial \psi_{l+1,j}}{\partial b_{l,j}} = z_{l+1,j}(1 - z_{l+1,j}) \\ \frac{\partial z_{l+1,j}}{\partial w_{l,i,j}} = z_{l+1,j}(1 - z_{l+1,j}) z_{l,i} = z_{l,i} \cdot \frac{\partial z_{l+1,j}}{\partial b_{l,j}} \\ \frac{\partial z_{l+1,j}}{\partial z_{l,i}} = z_{l+1,j}(1 - z_{l+1,j}) w_{l,i,j} = w_{l,i,j} \cdot \frac{\partial z_{l+1,j}}{\partial b_{l,j}} \end{cases}. \tag{10}$$

Let us define

$$E \triangleq \frac{1}{2} \sum_{j=1}^{n_y} (y_j - t_j)^2, \tag{11}$$

we have

$$\frac{\partial E}{\partial b_{h+1,j}} = \frac{\partial E}{\partial y_j} \cdot \frac{\partial y_j}{\partial b_{h+1,j}} = (y_j - t_j) \cdot y_j(1 - y_j). \tag{12}$$

For $l \leq h$,

$$\begin{aligned} \frac{\partial E}{\partial b_{l,j}} &= \frac{\partial E}{\partial z_{l+1,j}} \cdot \frac{\partial z_{l+1,j}}{\partial b_{l,j}} \\ &= \frac{\partial E}{\partial z_{l+1,j}} \cdot z_{l+1,j}(1 - z_{l+1,j}). \end{aligned} \tag{13}$$

Further

$$\begin{aligned} \frac{\partial E}{\partial z_{l+1,j}} &= \sum_{k=1}^{m_{l+2}} \left(\frac{\partial E}{\partial z_{l+2,k}} \cdot \frac{\partial z_{l+2,k}}{\partial z_{l+1,j}} \right) \\ &= \sum_{k=1}^{m_{l+2}} \left(\frac{\partial E}{\partial z_{l+2,k}} \cdot \frac{\partial z_{l+2,k}}{\partial b_{l+1,k}} \cdot w_{l+1,j,k} \right) \\ &= \sum_{k=1}^{m_{l+2}} \left(\frac{\partial E}{\partial b_{l+1,k}} \cdot w_{l+1,j,k} \right). \end{aligned} \tag{14}$$

Finally,

$$\frac{\partial E}{\partial b_{l,j}} = z_{l+1,j}(1 - z_{l+1,j}) \cdot \sum_{k=1}^{m_{l+2}} \left(\frac{\partial E}{\partial b_{l+1,k}} \cdot w_{l+1,j,k} \right). \tag{15}$$

The complexity to compute $\frac{\partial E}{\partial b_{h+1,j}}$ is $O(1)$, and the complexity to compute $\frac{\partial E}{\partial b_{l,j}}$, where $l \in [1:h]$, is $O(m_{l+2})$. Hence, the complexity to compute $\frac{\partial E}{\partial \mathbf{b}_{h+1}}$ is $O(m_{h+2})$, and the complexity to compute $\frac{\partial E}{\partial \mathbf{b}_l}$, where $l \in [1:h]$, is $O(m_{l+1}m_{l+2})$. Further, the additional complexity to compute $\frac{\partial E}{\partial \mathbf{W}_l}$, where $l \in [1:(h+1)]$, is $O(m_l m_{l+1})$.

III. RECTIFIER

For $l \in [1:(h+1)]$ and $j \in [1:m_{l+1}]$, we define

$$\psi_{l+1,j} \triangleq b_{l,j} + \sum_{i=1}^{m_l} w_{l,i,j} z_{l,i}. \tag{16}$$

Then $z_{l+1,j} = \varphi(\psi_{l+1,j})$. It is easy to get

$$\frac{\partial z_{l+1,j}}{\partial b_{l,j}} = \frac{\partial \varphi(\psi_{l+1,j})}{\partial \psi_{l+1,j}} \cdot \frac{\partial \psi_{l+1,j}}{\partial b_{l,j}} = \mathbf{1}_{z_{l+1,j}>0}. \tag{17}$$

Then

$$\begin{cases} \frac{\partial z_{l+1,j}}{\partial w_{l,i,j}} = z_{l,i} \cdot \frac{\partial z_{l+1,j}}{\partial b_{l,j}} \\ \frac{\partial z_{l+1,j}}{\partial z_{l,i}} = w_{l,i,j} \cdot \frac{\partial z_{l+1,j}}{\partial b_{l,j}} \end{cases}. \tag{18}$$

We have

$$\frac{\partial E}{\partial b_{h+1,j}} = \frac{\partial E}{\partial y_j} \cdot \frac{\partial y_j}{\partial b_{h+1,j}} = (y_j - t_j) \cdot \mathbf{1}_{y_j>0}. \tag{19}$$

For $l \leq h$,

$$\frac{\partial E}{\partial b_{l,j}} = \frac{\partial z_{l+1,j}}{\partial b_{l,j}} \cdot \sum_{k=1}^{m_{l+2}} \left(\frac{\partial E}{\partial b_{l+1,k}} \cdot w_{l+1,j,k} \right) = \mathbf{1}_{z_{l+1,j}>0} \cdot \sum_{k=1}^{m_{l+2}} \left(\frac{\partial E}{\partial b_{l+1,k}} \cdot w_{l+1,j,k} \right). \tag{20}$$

IV. CONVOLUTIONAL NEURAL NETWORK

Consider an ANN including $(h+2)$ layers. The 1-th layer is the input layer; the $(h+2)$-th layer is the output layer; and the intermediate h layers are hidden layers.

The l-th layer, where $l \in [1:(h+2)]$, includes m_l nodes. Especially, the input layer includes $m_1 = n_x$ input nodes and the output layer includes $m_{h+2} = n_y$ output nodes.

The i-th node of the l-th layer is denoted by $z_{l,i}$, where $l \in [1:(h+2)]$ and $i \in [1:m_l]$. Especially, $z_{1,i} = x_i$ is the i-th input node and $z_{h+2,i} = y_i$ is the i-th output node. The l-th layer node vector is denoted by $\mathbf{z}_l \in \mathbb{R}^{m_l}$. Especially, $\mathbf{z}_1 = \mathbf{x} \in \mathbb{R}^{n_x}$ and $\mathbf{z}_{h+2} = \mathbf{y} \in \mathbb{R}^{n_y}$.

Let $w_{l,i,j}$ denote the weight of the connection between the i-th node of the l-th layer and the j-th node of the $(l+1)$-th layer, where $l \in [1:(h+1)]$, $i \in [1:m_l]$, and $j \in [1:m_{l+1}]$. The l-th layer weight matrix is denoted by $\mathbf{W}_l \in \mathbb{R}^{m_l \times m_{l+1}}$.

Let $b_{l,j}$ denote the bias associated with the j-th node of the $(l+1)$-th layer, where $l \in [1:(h+1)]$ and $j \in [1:m_{l+1}]$. The l-th bias vector is denoted by $\mathbf{b}_l \in \mathbb{R}^{m_{l+1}}$.

Assume that the l-th layer is a convolutional layer with a length-K_l kernel, where K_l is an odd. Obviously, $m_{l+1} = (m_l - K_l) + 1$. Let $\boldsymbol{\omega}_l \triangleq (\omega_{l,1} \cdots, \omega_{l,K_l})$. Then

- For $j \le i \le (j + K_l - 1)$, $w_{l,i,j} = \omega_{l,i-j+1}$; and
- For $\max(1, i - K_l + 1) \le j \le \min(i, m_{l+1})$, $w_{l,i,j} = \omega_{l,i-j+1}$.

For $l \in [1:(h+1)]$ and $j \in [1:m_{l+1}]$, we define

$$\psi_{l+1,j} \triangleq b_{l,j} + \sum_{i=1}^{m_l} w_{l,i,j} z_{l,i} = b_{l,j} + \sum_{i=j}^{j+K_l-1} w_{l,i,j} z_{l,i} = b_{l,j} + \sum_{i'=1}^{K_l} \omega_{l,i'} z_{l,(j-1)+i'}. \tag{21}$$

Then $z_{l+1,j} = \varphi(\psi_{l+1,j})$. It is easy to get

$$\frac{\partial z_{l+1,j}}{\partial b_{l,j}} = \frac{\partial \varphi(\psi_{l+1,j})}{\partial \psi_{l+1,j}} \cdot \frac{\partial \psi_{l+1,j}}{\partial b_{l,j}} = \frac{\partial \varphi(\psi_{l+1,j})}{\partial \psi_{l+1,j}} = \mathbf{1}_{z_{l+1,j}>0} \tag{22}$$

and

$$\frac{\partial z_{l+1,j}}{\partial \omega_{l,i'}} = z_{l,(j-1)+i'} \cdot \frac{\partial z_{l+1,j}}{\partial b_{l,j}}. \tag{23}$$

Finally, we have

$$\frac{\partial E}{\partial b_{l,j}} = \frac{\partial E}{\partial z_{l+1,j}} \cdot \frac{\partial z_{l+1,j}}{\partial b_{l,j}} \tag{24}$$

and

$$\frac{\partial E}{\partial \omega_{l,i'}} = \sum_{j=1}^{m_{l+1}} \frac{\partial E}{\partial z_{l+1,j}} \cdot \frac{\partial z_{l+1,j}}{\partial \omega_{l,i'}} = \sum_{j=1}^{m_{l+1}} z_{l,(j-1)+i'} \cdot \frac{\partial E}{\partial b_{l,j}}, \tag{25}$$

where $\frac{\partial E}{\partial z_{h+2,j}} = z_{h+2,j} - t_j$ and for $l \leq h$,

$$\frac{\partial E}{\partial z_{l+1,j}} = \sum_{k=1}^{m_{l+2}} \left(\frac{\partial E}{\partial b_{l+1,k}} \cdot w_{l+1,j,k} \right). \tag{26}$$

If the $(l+1)$-th layer is a convolutional layer, then $m_{l+2} = (m_{l+1} - K_{l+1}) + 1$ and

$$\begin{aligned} \frac{\partial E}{\partial z_{l+1,j}} &= \sum_{k=\max(1,j-K_{l+1}+1)}^{\min(j,m_{l+2})} \left(\frac{\partial E}{\partial b_{l+1,k}} \cdot w_{l+1,j,k} \right) \\ &= \sum_{k=\max(1,j-K_{l+1}+1)}^{\min(j,m_{l+2})} \left(\frac{\partial E}{\partial b_{l+1,k}} \cdot \omega_{l+1,j-k+1} \right). \end{aligned} \tag{27}$$

V. 2D Convolutional Neural Network

Consider an ANN including $(h+2)$ layers. The 1-th layer is the input layer; the $(h+2)$-th layer is the output layer; and the intermediate h layers are hidden layers. The l-th layer, where $l \in [1:(h+2)]$, includes $m_l \times n_l$ nodes. The (i_x, i_y)-th node of the l-th layer is denoted by $z_{l,(i_x,i_y)}$, where $l \in [1:(h+2)]$ and $(i_x, i_y) \in [1:m_l] \times [1:n_l]$. Let $w_{l,(i_x,i_y),(j_x,j_y)}$ denote the weight of the connection between the (i_x, i_y)-th node of the l-th layer and the (j_x, j_y)-th node of the $(l+1)$-th layer. Let $b_{l,(j_x,j_y)}$ denote the bias associated with the (j_x, j_y)-th node of the $(l+1)$-th layer.

Assume that the l-th layer is a convolutional layer with a length-K_l kernel, where K_l is an odd. Obviously, $m_{l+1} = (m_l - K_l) + 1$ and $n_{l+1} = (n_l - K_l) + 1$. Let

$$\mathbf{\Omega}_l = \begin{pmatrix} \omega_{l,(1,1)} & \cdots & \omega_{l,(1,K_l)} \\ \vdots & \ddots & \vdots \\ \omega_{l,(K_l,1)} & \cdots & \omega_{l,(K_l,K_l)} \end{pmatrix} \in \mathbb{R}^{K_l \times K_l}. \tag{28}$$

Then

$$w_{l,(i_x,i_y),(j_x,j_y)} = \omega_{l,(i_x-j_x+1,i_y-j_y+1)}, \tag{29}$$

where

- $j_x \leq i_x \leq (j_x + K_l - 1)$ and $j_y \leq i_y \leq (j_y + K_l - 1)$; and
- $\max(1, i_x - K_l + 1) \leq j_x \leq \min(i_x, m_{l+1})$ and $\max(1, i_y - K_l + 1) \leq j_y \leq \min(i_y, n_{l+1})$.

We define

$$\psi_{l+1,(j_x,j_y)} \triangleq b_{l,(j_x,j_y)} + \sum_{i'_x=1}^{K_l} \sum_{i'_y=1}^{K_l} \omega_{l,(i'_x,i'_y)} z_{l,(i_x,i_y)}, \tag{30}$$

where $i'_x = i_x - j_x + 1$ and $i'_y = i_y - j_y + 1$. Then $z_{l+1,(j_x,j_y)} = \varphi(\psi_{l+1,(j_x,j_y)})$. Easy to get

$$\begin{cases} \frac{\partial z_{l+1,(j_x,j_y)}}{\partial b_{l,(j_x,j_y)}} = \mathbf{1}_{z_{l+1,(j_x,j_y)}>0} \\ \frac{\partial z_{l+1,(j_x,j_y)}}{\partial \omega_{l,(i'_x,i'_y)}} = z_{l,(i_x,i_y)} \cdot \frac{\partial z_{l+1,(j_x,j_y)}}{\partial b_{l,(j_x,j_y)}} = z_{l,(i_x,i_y)} \cdot \mathbf{1}_{z_{l+1,(j_x,j_y)}>0} \end{cases}. \tag{31}$$

Finally, we have

$$\begin{cases} \frac{\partial E}{\partial b_{l,(j_x,j_y)}} = \frac{\partial E}{\partial z_{l+1,(j_x,j_y)}} \cdot \frac{\partial z_{l+1,(j_x,j_y)}}{\partial b_{l,(j_x,j_y)}} \\ \frac{\partial E}{\partial \omega_{l,(i'_x,i'_y)}} = \sum_{j_x=1}^{m_{l+1}} \sum_{j_y=1}^{n_{l+1}} \frac{\partial E}{\partial z_{l+1,(j_x,j_y)}} \cdot \frac{\partial z_{l+1,(j_x,j_y)}}{\partial \omega_{l,(i'_x,i'_y)}} \end{cases}, \tag{32}$$

where

$$\frac{\partial E}{\partial z_{l+1,(j_x,j_y)}} = \sum_{k_x=1}^{m_{l+2}} \sum_{k_y=1}^{n_{l+2}} \left(\frac{\partial E}{\partial b_{l+1,(k_x,k_y)}} \cdot w_{l+1,(j_x,j_y),(k_x,k_y)} \right). \tag{33}$$

If the $(l+1)$-th layer is a convolutional layer, then $m_{l+2}=(m_{l+1}-K_{l+1})+1$, $n_{l+2}=(n_{l+1}-K_{l+1})+1$, and

$$\frac{\partial E}{\partial z_{l+1,(j_x,j_y)}}=\sum_{k_x=\max(1,j_x-K_{l+1}+1)}^{\min(j_x,m_{l+2})}\sum_{k_y=\max(1,j_y-K_{l+1}+1)}^{\min(j_y,n_{l+2})}\left(\frac{\partial E}{\partial b_{l+1,(k_x,k_y)}}\cdot\omega_{l+1,(j_x-k_x+1,j_y-k_y+1)}\right). \tag{34}$$

VI. Restricted Boltzmann Machine

Assume there are m visible nodes and n hidden nodes. Let v_i denote the i-th visible node, where $i\in[1:m]$, and z_j the j-th hidden node, where $j\in[1:n]$. Let $\mathbf{v}=(v_1,\cdots,v_m)^\top\in\mathbb{B}^m$ and $\mathbf{h}=(h_1,\cdots,h_n)^\top\in\mathbb{B}^n$. Let

$$\mathbf{W}=\begin{pmatrix} w_{1,1}, & \cdots, & w_{1,n}\\ \vdots & \ddots & \vdots\\ w_{m,1}, & \cdots, & w_{m,n}\end{pmatrix}\in\mathbb{R}^{m\times n}. \tag{35}$$

Let us define the energy with respect to $\mathbf{v}$ and $\mathbf{h}$ as

$$E(\mathbf{v},\mathbf{h})\triangleq-\mathbf{a}^\top\mathbf{v}-\mathbf{b}^\top\mathbf{h}-\mathbf{v}^\top\mathbf{W}\mathbf{h}. \tag{36}$$

and the probability of $(\mathbf{v},\mathbf{h})$ as

$$p(\mathbf{v},\mathbf{h})\propto e^{-E(\mathbf{v},\mathbf{h})}=e^{\mathbf{a}^\top\mathbf{v}+\mathbf{b}^\top\mathbf{h}+\mathbf{v}^\top\mathbf{W}\mathbf{h}}. \tag{37}$$

Hence, we have

$$p(v_i,\mathbf{h})\propto e^{-E(v_i,\mathbf{h})}=e^{a_iv_i+\mathbf{b}^\top\mathbf{h}+v_i\sum_{j=1}^n w_{i,j}z_j}=e^{\mathbf{b}^\top\mathbf{h}+v_i(a_i+\sum_{j=1}^n w_{i,j}z_j)} \tag{38}$$

and

$$p(z_j,\mathbf{v})\propto e^{-E(z_j,\mathbf{v})}=e^{b_jz_j+\mathbf{a}^\top\mathbf{v}+z_j\sum_{i=1}^m w_{i,j}v_i}=e^{\mathbf{a}^\top\mathbf{v}+z_j(b_j+\sum_{i=1}^m w_{i,j}v_i)} \tag{39}$$

Then

$$\begin{aligned} p(v_i = 1|\mathbf{h}) &= \frac{p(v_i = 1, \mathbf{h})}{p(v_i = 1, \mathbf{h}) + p(v_i = 0, \mathbf{h})} \\ &= \frac{e^{\mathbf{b}^\top \mathbf{h} + (a_i + \sum_{j=1}^n w_{i,j} z_j)}}{e^{\mathbf{b}^\top \mathbf{h}} + e^{\mathbf{b}^\top \mathbf{h} + (a_i + \sum_{j=1}^n w_{i,j} z_j)}} \\ &= \frac{e^{(a_i + \sum_{j=1}^n w_{i,j} z_j)}}{1 + e^{(a_i + \sum_{j=1}^n w_{i,j} z_j)}} \\ &= \frac{1}{1 + e^{-(a_i + \sum_{j=1}^n w_{i,j} z_j)}} \\ &= \psi\left(a_i + \sum_{j=1}^n w_{i,j} z_j\right). \end{aligned} \tag{40}$$

Similarly,

$$p(z_j = 1|\mathbf{v}) = \psi\left(b_j + \sum_{i=1}^m w_{i,j} v_i\right). \tag{41}$$

$$\frac{\partial \log p(\mathbf{v}, \mathbf{h})}{\partial w_{i,j}} = \frac{\partial \mathbf{v}^\top \mathbf{W} \mathbf{h}}{\partial w_{i,j}} = v_i z_j. \tag{42}$$

$$p(\mathbf{v}) = \sum_{\mathbf{h} \in \mathbb{B}^n} p(\mathbf{v}, \mathbf{h}) \tag{43}$$

$$\frac{\partial \log p(\mathbf{v})}{\partial w_{i,j}} = \tag{44}$$

Assume that the l-th layer is a convolutional layer with a length-K_l kernel and stride is S_l. Obviously,

$$m_{l+1} = (m_l - K_l)/S_l + 1. \tag{45}$$

It is easy to get

$$w_{l,i,j} = \begin{cases} 0, & i \notin \{(j-1)S_l + [1:K_l]\} \\ \omega_{l,i'}, & i \in \{(j-1)S_l + [1:K_l]\} \end{cases}, \tag{46}$$

where $i' = i - (j-1)S_l \in [1:K_l]$.

For $l \in [1:(h+1)]$ and $j \in [1:m_{l+1}]$, we define

$$\psi_{l+1,j} \triangleq b_{l,j} + \sum_{i=1}^{m_l} w_{l,i,j} z_{l,i} = b_{l,j} + \sum_{i=(j-1)S_l+1}^{(j-1)S_l+K_l} w_{l,i,j} z_{l,i} = b_{l,j} + \sum_{i'=1}^{K_l} \omega_{l,i'} z_{l,(j-1)S_l+i'}. \tag{47}$$

Then $z_{l+1,j} = \varphi(\psi_{l+1,j})$. It is easy to get It is easy to get

$$\frac{\partial z_{l+1,j}}{\partial b_{l,j}} = \frac{\partial \varphi(\psi_{l+1,j})}{\partial \psi_{l+1,j}} \cdot \frac{\partial \psi_{l+1,j}}{\partial b_{l,j}} = \frac{\partial \varphi(\psi_{l+1,j})}{\partial \psi_{l+1,j}} = \mathbf{1}_{z_{l+1,j}>0}. \tag{48}$$

Hence,

$$\frac{\partial z_{l+1,j}}{\partial \omega_{l,i'}} = z_{l,(j-1)S_l+i'} \cdot \frac{\partial z_{l+1,j}}{\partial b_{l,j}} \tag{49}$$

Then

$$\frac{\partial E}{\partial \omega_{l,i'}} = \sum_{j=1}^{m_{l+1}} \frac{\partial E}{\partial z_{l+1,j}} \cdot \frac{\partial z_{l+1,j}}{\partial \omega_{l,i'}}, \tag{50}$$

where

$$\frac{\partial E}{\partial z_{l+1,j}} = \sum_{k=1}^{m_{l+2}} \left(\frac{\partial E}{\partial b_{l+1,k}} \cdot w_{l+1,j,k} \right). \tag{51}$$

Finally,

$$\frac{\partial E}{\partial b_{l,j}} = \frac{\partial z_{l+1,j}}{\partial b_{l,j}} \cdot \sum_{k=1}^{m_{l+2}} \left(\frac{\partial E}{\partial b_{l+1,k}} \cdot w_{l+1,j,k} \right). \tag{52}$$

If the $(l+1)$-th layer is a convolutional layer, then

$$m_{l+2} = (m_{l+1} - K_{l+1})/S_{l+1} + 1 \tag{53}$$

$w_{l+1,j,k} = \omega_{l+1,j'}$, where

$$j' = j - (k-1)S_{k+1} \in [1 : K_{l+1}]. \tag{54}$$

An autoencoder can be taken as an Artificial Neural Network (ANN) including 1 input layer, 1 hidden layer, and 1 output layer. The input layer includes m input nodes. The output layer includes m output nodes. The hidden layer includes n hidden nodes. Denote the i-th input node as x_i, the i-th output node as y_i, and the j-th hidden node as z_j, where $i \in [1 : m]$ and $j \in [1 : n]$. Let $\mathbf{W} \in \mathbb{R}^{m \times n}$ be the weight matrix and $\mathbf{b} \in \mathbb{R}^n$ be the bias vector for the encoding. Let $\mathbf{W}^\top$ be the weight matrix and $\hat{\mathbf{b}} \in \mathbb{R}^m$ be the bias vector for the decoding. Thus

$$z_j = \psi(b_j + \sum_{i'=1}^{k} w_{i'} x_{i'+s(j-1)}). \tag{55}$$

It is easy to get $\frac{\partial z_j}{\partial b_j} = z_j(1 - z_j)$ and $\frac{\partial z_j}{\partial w_{i,j}} = x_i \cdot \frac{\partial z_j}{\partial b_j}$. Further,

$$y_i = \psi(\hat{b}_i + \sum_{j=1}^{n} w_{i,j} z_j). \tag{56}$$

It is easy to get $\frac{\partial y_i}{\partial \hat{b}_i} = y_i(1 - y_i)$ and $\frac{\partial y_i}{\partial z_j} = w_{i,j} \cdot \frac{\partial y_i}{\partial \hat{b}_i}$. In addition,

$$\frac{\partial y_i}{\partial w_{i,j}} = \frac{\partial y_i}{\partial \hat{b}_i} \cdot \left(z_j + w_{i,j} \cdot \frac{\partial z_j}{\partial w_{i,j}} \right) \tag{57}$$

and for $i' \neq i$,

$$\frac{\partial y_{i'}}{\partial w_{i,j}} = \frac{\partial y_{i'}}{\partial \hat{b}_{i'}} \cdot \left(w_{i',j} \cdot \frac{\partial z_j}{\partial w_{i,j}} \right) \tag{58}$$

The error is

$$E = \frac{1}{2}\sum_{i=1}^{m}(y_i - x_i)^2 \tag{59}$$

Hence,

$$\frac{\partial E}{\partial \hat{b}_i} = \frac{\partial E}{\partial y_i} \cdot \frac{\partial y_i}{\partial \hat{b}_i} = (y_i - x_i)y_i(1 - y_i). \tag{60}$$

$n = \frac{m-k}{s} + 1$.

Let x^m be the input.

Filter Stride Pad Pooling