# Mathematical Foundation of DNN : HW 7

Jeong Min Lee

April 29, 2024

## 1

Let $x^\star = \max\{x_1, \cdots, x_n\}$.

### a

$$\begin{aligned}
\lim_{\beta\to\infty} \frac{1}{\beta} \log\left(\sum_{i=1}^{n} \exp(\beta x_i)\right) &= \lim_{\beta\to\infty} \frac{1}{\beta} \log\left(\exp(\beta x^\star) \sum_{i=1}^{n} \exp(\beta(x_i - x^\star))\right) \\
&= \lim_{\beta\to\infty} \frac{1}{\beta} \log(\exp(\beta x^\star)) + \frac{1}{\beta} \log\left(\sum_{i} \exp(\beta(x_i - x^\star))\right) \\
&= x^\star
\end{aligned}$$

The last equation is held since the second term in second equality becomes zero. This can be proved as follow.

$$0 \le \lim_{\beta\to\infty} \frac{1}{\beta} \log\left(\sum_{i=1}^{n} \exp\left(\beta(x_i - x^\star)\right)\right) \le \lim_{\beta\to\infty} \frac{1}{\beta} \log\left(\sum_{i=1}^{n} 1\right) = \lim_{\beta\to\infty} \frac{\log n}{\beta} = 0$$

### b

$$\begin{aligned}
\left[\nabla \nu_1\right]_j &= \frac{\partial}{\partial x_j} \log\left(\sum_{i=1}^{n} \exp(x_i)\right) \\
&= \frac{\sum_{i=1}^{n} \exp(x_i)\delta_{ij}}{\sum_{i=1}^{n} \exp(x_i)} \\
&= \frac{\exp(x_j)}{\sum_{i=1}^{n} \exp(x_i)} \\
&= \mu_j
\end{aligned}$$

### c

$$\begin{aligned}
\left[\nabla \nu_\beta\right]_j &= \frac{1}{\beta} \frac{\partial}{\partial x_j} \log\left(\sum_{i=1}^{n} \exp(\beta x_i)\right) \\
&= \frac{1}{\beta} \frac{\beta \exp(\beta x_j)}{\sum_{i=1}^{n} \exp(\beta x_i)} \\
&= \frac{\exp(\beta x_j)}{\sum_{i=1}^{n} \exp(\beta x_i)}
\end{aligned}$$

$$\begin{aligned}
\lim_{\beta\to\infty} \nabla \nu_\beta &= \lim_{\beta\to\infty} \frac{1}{\sum_{i=1}^{n} \exp(\beta x_i)} \begin{pmatrix} \exp(\beta x_1) \\ \vdots \\ \exp(\beta x_n) \end{pmatrix} \\
&= \lim_{\beta\to\infty} \frac{\exp(\beta x_{i_{max}})}{\sum_{i=1}^{n} \exp(\beta x_i)} \begin{pmatrix} \exp(\beta(x_1 - x_{i_{max}})) \\ \vdots \\ \exp(\beta(x_n - x_{i_{max}})) \end{pmatrix} \\
&= \lim_{\beta\to\infty} \frac{1}{\sum_{i=1}^{n} \exp(\beta(x_i - x_{i_{max}}))} \begin{pmatrix} \exp(\beta(x_1 - x_{i_{max}})) \\ \vdots \\ \exp(\beta(x_n - x_{i_{max}})) \end{pmatrix}
\end{aligned}$$

To prove this, the following equality is useful.

$$\lim_{\beta\to\infty}\sum_{i=1}^{n}\exp\left(\beta(x_i - x_{i_{max}})\right) = \sum_{i=1}^{n}\lim_{\beta\to\infty}\exp\left(\beta(x_i - x_{i_{max}})\right) = \begin{cases}0 \text{ if } i \neq i_{max}\\ 1 \text{ if } i = i_{max}\end{cases}$$

Since $!\exists i_{max} = \arg\max_{i\leq i\leq n}\{x_1,\cdots,x_n\}$,

$$\begin{aligned}&\lim_{\beta\to\infty}\frac{1}{\sum_{i=1}^{n}\exp(\beta(x_i - x_{i_{max}}))} = 1\\ &\begin{pmatrix}\exp(\beta(x_1 - x_{i_{max}}))\\ \vdots\\ \exp(\beta(x_n - x_{i_{max}}))\end{pmatrix} = \begin{pmatrix}0 & \cdots & 1 & \cdots & 0\end{pmatrix}^T = e_{i_{max}}\end{aligned}$$

Thus, $\lim_{\beta\to\infty}\nabla\nu_\beta = e_{i_{max}}$

## 2

First of all, consider a convolution layer. Looking at the convolution operation for each patch of input feature, the number of multiplication and addition are same to $C_{in}k^2$. For each operation, we have $C_{out}H_{out}W_{out}$ number of output features. Thus, by multiplying them, one can derive the number of addition and multiplication for each forward pass. Next, consider the linear layer. It is a special case of convolution layer where $k = 1, H_{out} = 1, W_{out} = 1$. Thus, the number of addition and multiplication in linear layer is $(number\ of\ input\ feature) \times (number\ of\ output\ feature)$. Noting the formulae above, the total number of addition and multiplication in linear layer is $2\times(3\times 64\times 11^2\times 55^2 + 64\times 192\times 5^2\times 27^2 + 192\times 384\times 3^2\times 13^2 + 384\times 256\times 3^2\times 13^2 + 256\times 256\times 3^2\times 13^2) = 117,243,904$ , while in convolution layer, it is $2\times\left(256\times 6^2\times 4096 + 4096\times 4096 + 4096\times 1000\right) = 1,311,133,056$.

## 4

### a

$$\frac{\partial y_L}{\partial y_{L-1}} = \frac{\partial}{\partial y_{L-1}}(A_{w_L}y_{L-1} + b_L\mathbf{1}_{n_L}) = A_{w_L}$$

$$\begin{aligned}\left(\frac{\partial y_l}{\partial y_{l-1}}\right)^{(ij)} &= \frac{\partial y_l^{(i)}}{\partial y_{l-1}^{(j)}}\\ &= \frac{\partial}{\partial y_{l-1}^{(j)}}\sigma\left(\sum_k A_{w_l}^{(ik)}y_{l-1}^{(k)} + b_l\right)\\ &= \sigma'\left(\sum_k A_{w_l}^{(ik)}y_{l-1}^{(k)} + b_l\right)\left(\sum_k \frac{\partial}{\partial y_{l-1}^{(i)}}A_{w_l}^{(ik)}y_{l-1}^{(k)}\right)\\ &= \sigma'\left(\sum_k A_{w_l}^{(ik)}y_{l-1}^{(k)} + b_l\right)A_{w_l}^{(ij)}\end{aligned}$$

Note that $\partial y_l/\partial y_{l-1}{}^{(ij)} = \sum_k \operatorname{diag}\left(\sigma'\left(A_{w_l}y_{l-1} + b_l\mathbf{1}_{n_l}\right)\right)^{(ik)}A_{w_l}^{(kj)} = \sigma'\left(A_{w_l}y_{l-1} + b_l\mathbf{1}_{n_l}\right)^{(i)}A_{w_l}^{(ij)}$ $= \sigma'\left(A_{w_l}^{(ik)}y_{l-1}^{(k)} + b_l\right)A_{w_l}^{(kj)}$. Thus, proof is done. To proof furthermore, we have to be aware of the following property.

$$(A_{w_l}y_{l-1})^{(i)} = \sum_{k=1}^{n_{l-1}} A_{w_l}^{(ik)}y_{l-1}^{(k)} = \sum_{k=i}^{i+f_l-1} w_l^{(k-i+1)}y_{l-1}^{(k)}$$

Note that $A_{w_l}^{(i,j)} = w_l^{(j)}$ if $i \le j \le i + f_l - 1$ otherwise 0.

$$
\begin{aligned}
\left(\frac{\partial y_L}{\partial w_l}\right)^{(i)} &= \sum_{j=1}^{n_l} \frac{\partial y_L}{\partial y_l^{(j)}} \frac{\partial y_l^{(j)}}{\partial w_l^{(i)}} \\
&= \sum_{j=1}^{n_l} \frac{\partial y_L}{\partial y_l^{(j)}} \frac{\partial}{\partial w_l^{(i)}} \sigma\left(\sum_k A_{w_l}^{(jk)} y_{l-1}^{(k)} + b_l\right) \\
&= \sum_{j=1}^{n_l} \frac{\partial y_L}{\partial y_l^{(j)}} \sigma'\left(\sum_k A_{w_l}^{(jk)} y_{l-1}^{(k)} + b_l\right) \left(\sum_k \frac{\partial}{\partial w_l^{(i)}} A_{w_l}^{(jk)} y_{l-1}^{(k)}\right) \\
&= \sum_{j=1}^{n_l} \frac{\partial y_L}{\partial y_l^{(j)}} \sigma'\left(\sum_k A_{w_l}^{(jk)} y_{l-1}^{(k)} + b_l\right) \left(\sum_{k=j}^{j+f_l-1} \frac{\partial}{\partial w_l^{(i)}} w_l^{(k)} y_{l-1}^{(k)}\right) \\
&= \sum_{j=1}^{n_l} \frac{\partial y_L}{\partial y_l^{(j)}} \sigma'\left(\sum_k A_{w_l}^{(jk)} y_{l-1}^{(k)} + b_l\right) y_{l-1}^{(i+j-1)} \\
&= \sum_{j=1}^{n_l} v_l^{(j)} y_{l-1}^{(i+j-1)}
\end{aligned}
$$

Note that $v_l^{(j)} = \partial y_L / \partial y_l^{(j)} \sigma'\left(\sum_k A_{w_l}^{(jk)} y_{l-1}^{(k)} + b_l\right)$ and $\left(C_{v_l^T} y_{l-1}\right)^{(i)} = \sum_k \left(C_{v_l^T}^{(ik)}\right) y_{l-1}^{(k)} = \sum_{k=i}^{i+n_l-1} v_l^{(k-i+1)} y_{l-1}^{(k)} = \sum_{j=1}^{n_l} v_l^{(j)} y_{l-1}^{(i+j-1)}$.

$$
\begin{aligned}
\frac{\partial y_L}{\partial b_l} &= \sum_{j=1}^{n_l} \frac{\partial y_L}{\partial y_l^{(j)}} \frac{\partial y_l^{(j)}}{\partial b_l} \\
&= \sum_{j=1}^{n_l} \frac{\partial y_L}{\partial y_l^{(j)}} \sigma'\left(\sum_k A_{w_l}^{(jk)} y_{l-1}^{(k)} + b_l\right) \frac{\partial}{\partial b_l} \sum_k A_{w_l}^{(jk)} y_{l-1}^{(k)} + b_l \\
&= \sum_{j=1}^{n_l} v_l^{(j)} \\
&= v_l \cdot \mathbf{1}_{n_l}
\end{aligned}
$$

## b

In the context of convolutional neural networks (CNNs), where operations like convolutions are prevalent, the traditional matrix-vector multiplication may not be the most efficient approach. Instead, we can leverage specific matrix-vector or vector-matrix products with respect to $A_{w_i}$ or $A_{w_i}^T$ to streamline the forward pass and backpropagation. Instead of forming the full matrix $A_{w_i}$, which represents the convolution with filter $w_i$, we can directly apply the convolution operation to the input data and the filter $w_i$. This approach eliminates the need to explicitly construct the entire matrix and significantly reduces computational overhead. By utilizing specialized convolutional operations tailored to the network architecture, we can achieve substantial improvements in computational efficiency during the forward pass. Similarly, in backpropagation, we can efficiently calculate the gradients. Refering to problem (a), we can avoid direct matrix calculation process by performing convolution and transpose convolution. By utilizing these tailored approaches for matrix-vector or vector-matrix products with respect to convolutional operations, we can significantly optimize both the forward pass and backpropagation in CNNs, leading to improved training efficiency and overall performance of the network.