MathDNN Homework 5

Department of Computer Science and Engineering
2021-16988 Jaewan Park

Problem 2

When $i \neq L$, we can calculate the following.

$$\begin{aligned}
\frac{\partial y_L}{\partial b_i} &= \frac{\partial y_L}{\partial y_i}\frac{\partial y_i}{\partial b_i} \\
&= \frac{\partial y_L}{\partial y_{L-1}}\frac{\partial y_{L-1}}{\partial y_{L-2}}\cdots\frac{\partial y_{i+1}}{\partial y_i}\frac{\partial y_i}{\partial b_i} \\
&= A_L(\operatorname{diag}(\sigma'(\tilde{y_{L-1}}))A_{L-1})\cdots(\operatorname{diag}(\sigma'(\tilde{y_{i+1}}))A_{i+1})\operatorname{diag}(\sigma'(\tilde{y_i})) \\
\frac{\partial y_L}{\partial A_i} &= \operatorname{diag}(\sigma'(\tilde{y_i}))\left(\frac{\partial y_L}{\partial y_i}\right)^{\intercal} y_{i-1}^{\intercal} \\
&= \operatorname{diag}(\sigma'(\tilde{y_i}))A_{i+1}^{\intercal}\operatorname{diag}(\sigma'(\tilde{y_{i+1}}))^{\intercal}\cdots A_{L-1}^{\intercal}\operatorname{diag}(\sigma'(\tilde{y_{L-1}}))^{\intercal}A_L^{\intercal}y_{i-1}^{\intercal}
\end{aligned}$$

First, suppose A_j is small for some $j \in \{l+1, \cdots, L\}$. Then for any $i \in \{1, \cdots, l\}$, $i \neq L$ and A_j must exist among $A_{i+1}, \cdots, A_L$. Therefore one small matrix exists in the chain of matrix multiplication in the above calculations. Also, since $0 \leq \sigma'(z) \leq 0.25$ for all z where σ is the sigmoid activation function, all outputs of the function are relatively 'not too large' numbers and consequently $\operatorname{diag}(\sigma'(\tilde{y_k}))$ are all not too large matrices. Therefore the above calculations only contain not too large matrices and at least one small matrix, thus the results become small.

Next, suppose $\tilde{y_j}$ has large absolute value for some $j \in \{l+1, \cdots, L-1\}$. Then for any $i \in \{1, \cdots, l\}$, $i \neq L$ and at $\tilde{y_j}$ must exist among $\tilde{y_{i+1}}, \cdots, \tilde{y_{L-1}}$. Since $\sigma'(z) \to 0$ as $z \to \pm\infty$ where σ is the sigmoid activation function, $\operatorname{diag}(\sigma'(\tilde{y_j}))$ is absolutely 'small' as $\tilde{y_j}$ has absolute large value. Other values of A_k or $\operatorname{diag}(\sigma'(\tilde{y_k}))$ are all not too large. Therefore the above calculations only contain not too large matrices and at least one small matrix, thus the results become small.

Problem 3

To prevent confusion, notate the points calculated from Form I θ_{I}^k, and those from Form II θ_{II}^k.

Calculations of θ^1 from the two forms are identical.

$$\begin{aligned}
\theta_{\mathrm{I}}^1 &= \theta^0 - \alpha g^0 + \beta\big(\theta^0 - \theta^0\big) = \theta^0 - \alpha g^0 \\
\theta_{\mathrm{II}}^1 &= \theta^0 - \alpha v^1 = \theta^0 - \alpha\big(g^0 + \beta v^0\big) = \theta^0 - \alpha g^0
\end{aligned}$$

Now suppose calculations of $\theta^0, \cdots, \theta^k$ from the two forms are all identical. Then

$$\begin{aligned}\theta_{\mathrm{II}}^{k+1} &= \theta_{\mathrm{II}}^{k} - \alpha v^{k+1} \\ &= \theta_{\mathrm{I}}^{k} - \alpha\big(g^k + \beta v^k\big) = \theta_{\mathrm{I}}^{k} - \alpha g^k + \beta\big(-\alpha v^k\big) \\ &= \theta_{\mathrm{I}}^{k} - \alpha g^k + \beta\big(\theta_{\mathrm{II}}^{k} - \theta_{\mathrm{II}}^{k-1}\big) \\ &= \theta_{\mathrm{I}}^{k} - \alpha g^k + \beta\big(\theta_{\mathrm{I}}^{k} - \theta_{\mathrm{I}}^{k-1}\big) \\ &= \theta_{\mathrm{I}}^{k+1}.\end{aligned}$$

Therefore $\theta_{\mathrm{I}}^{k+1} = \theta_{\mathrm{II}}^{k+1}$, and by using mathematical induction, we can claim that Forms I and II produce the same $\theta^1, \theta^2, \cdots$ sequence.

Problem 4

Let the output of the first, third, and fourth convolutional layers y_4, y_5, y_6. Then $y_4[k,i,j]$ depends on $X[:, i-1:i+1, j-1:j+1]$, $y_1[k,i,j]$ depends on $y_4[:, i-1:i+1, j-1:j+1]$, $y_2[k,i,j]$ depends on $y_1[:, 2i-1:2i, 2j-1:2j]$, $y_5[k,i,j]$ depends on $y_2[:, i-1:i+1, j-1:j+1]$, $y_6[k,i,j]$ depends on $y_5[:, i-1:i+1, j-1:j+1]$, and $y_3[k,i,j]$ depends on $y_6[:, 2i-1:2i, 2j-1:2j]$.

As a result, $y_1[k,i,j]$ depends on $X[:, i-2:i+2, j-2:j+2]$, $y_2[k,i,j]$ depends on $X[:, 2i-3:2i+2; 2j-3:2j+2]$, and $y_3[k,i,j]$ depends on $X[:, 4i-9:4i+6, 4j-9:4j+6]$.

Problem 5

The number of trainable parameters between two layers can be calculated as $C_{\text{out}} \times (C_{\text{in}} \times F \times F + 1)$, where the addition of 1 is made due to the bias. The number of additions and multiplications are equal, both calculated as $C_{\text{out}} \times m_{\text{out}} \times n_{\text{out}} \times C_{\text{in}} \times F \times F$, where $m_{\text{out}} \times n_{\text{out}}$ is the output dimension. Additions are made between multiplied values, so the number of it should be one less than that of multiplications, but adding the bias makes them equal. The number of activation function evaluations can be calculated as $C_{\text{out}} \times m_{\text{out}} \times n_{\text{out}}$, since the function is applies after the convolution. Therefore the counts for each models are the following.

The First Model
Number of Trainable Parameters : $192 \times (256 \times 3 \times 3 + 1) + 96 \times (256 \times 5 \times 5 + 1) = 1057056$
Number of Additions : $192 \times 32 \times 32 \times 256 \times 3 \times 3 + 96 \times 32 \times 32 \times 256 \times 3 \times 3 = 679477248$
Number of Multiplications : $192 \times 32 \times 32 \times 256 \times 3 \times 3 + 96 \times 32 \times 32 \times 256 \times 3 \times 3 = 679477248$
Number of Activation Function Evaluations : $192 \times 32 \times 32 + 96 \times 32 \times 32 = 294912$

The Second Model
Number of Trainable Parameters : $64 \times (256 \times 1 \times 1 + 1) + 192 \times (64 \times 3 \times 3 + 1) + 64 \times (256 \times 1 \times 1 + 1) + 96 \times (64 \times 5 \times 5 + 1) = 297376$

Number of Additions : $64 \times 32 \times 32 \times 256 \times 1 \times 1 + 192 \times 32 \times 32 \times 64 \times 3 \times 3 + 64 \times 32 \times 32 \times 256 \times 1 \times 1 + 96 \times 32 \times 32 \times 64 \times 5 \times 5 = 304087040$
Number of Multiplications : $64 \times 32 \times 32 \times 256 \times 1 \times 1 + 192 \times 32 \times 32 \times 64 \times 3 \times 3 + 64 \times 32 \times 32 \times 256 \times 1 \times 1 + 96 \times 32 \times 32 \times 64 \times 5 \times 5 = 304087040$
Number of Activation Function Evaluations : $64 \times 32 \times 32 + 192 \times 32 \times 32 + 64 \times 32 \times 32 + 96 \times 32 \times 32 = 425984$

The second model has advantage in number of trainable parameters, while the number of additions, multiplications, and activation function evaluations are larger.