Submission for Deep Learning Exercise

Team: shallow_learning_group
Students: Batuhan Karaca

December 3, 2023

Pen and Paper tasks

Note that the boldface letter denotes vectors.

1)

With no padding, the formula for cross correlation is

$$\begin{aligned}
\hat{y}_i &= (\mathbf{w} \star \mathbf{x})_i + b \\
&= \sum_{l=1}^{|\mathbf{w}|} x_{s\cdot(i-1)+l} w_l + b \quad (1) \quad \text{where } |\mathbf{w}| \text{ is the size of the weights, and } s \text{ is the stride} \\
\hat{y}_1 &= \sum_{l=1}^{3} x_l w_l + b \\
&= x_1 w_1 + x_2 w_2 + x_3 w_3 + b \\
&= 14 \\
\hat{y}_2 &= \sum_{l=1}^{3} x_{l+1} w_l + b \\
&= x_2 w_1 + x_3 w_2 + x_4 w_3 + b \\
&= 20 \\
\hat{\mathbf{y}} &= \begin{bmatrix} 14 & 20 \end{bmatrix} \\
\mathcal{L}(\hat{y}, y) &= \frac{1}{2}[(14-2)^2 + (20-4)^2] \\
&= 200
\end{aligned}$$

2)

Note that $\mathbf{x_y}$ denotes the gradient of vector $\mathbf{x}$ with respect to $\mathbf{y}$, and $\star$ denotes cross correlation operator.

$$\begin{aligned}
\frac{\delta\mathcal{L}}{\delta w_i} &= \sum_{j=1}^{|\mathbf{y}|} \frac{\delta\mathcal{L}}{\delta\hat{y}_j}\frac{\delta\hat{y}_j}{\delta w_i} \\
&= \sum_{j=1}^{|\mathbf{y}|} (\hat{y}_j - y_j) x_{s\cdot(j-1)+i} \\
&= \sum_{j=1}^{|\mathbf{y}|} x_{(i-1)+j}(\hat{y}_j - y_j) \quad \text{due to the fact that } s = 1 \\
&= ((\hat{\mathbf{y}} - \mathbf{y}) \star \mathbf{x})_i \quad \text{from equation (1)} \\
\mathbf{L_w} &= (\hat{\mathbf{y}} - \mathbf{y}) \star \mathbf{x} \\
&= \mathbf{L}_{\hat{\mathbf{y}}} \star \mathbf{x} \\
&= \left(\begin{bmatrix}14 & 20\end{bmatrix} - \begin{bmatrix}2 & 4\end{bmatrix}\right) \star \begin{bmatrix}1 & 2 & 3 & 4\end{bmatrix} \\
&= \begin{bmatrix}12 & 16\end{bmatrix} \star \begin{bmatrix}1 & 2 & 3 & 4\end{bmatrix} \\
&= \begin{bmatrix}44 & 72 & 100\end{bmatrix} \\
\frac{\delta\mathcal{L}}{\delta b} &= \sum_{j=1}^{|\mathbf{y}|} \frac{\delta\mathcal{L}}{\delta\hat{y}_j}\frac{\delta\hat{y}_j}{\delta b} \\
&= \sum_{j=1}^{|\mathbf{y}|} (\hat{y}_j - y_j) \\
\mathbf{L}_b &= (\hat{\mathbf{y}} - \mathbf{y})\mathbf{1}_{1\times|\mathbf{y}|}^T \\
&= \mathbf{L}_{\hat{\mathbf{y}}}\mathbf{1}_{1\times|\mathbf{y}|}^T \\
&= 28
\end{aligned}$$

Please note that as derived above in general, the gradient for a weight parameter w_i is the dot product between the gradient of the next layer and a new array sampled from the layer input $\mathbf{x}$ in s intervals, starting from x_i. Trivially, when $s = 1$, this is cross-correlation with stride $s = 1$.

3)

Note that α is the learning rate.

$$\begin{aligned}
\mathbf{w}^t &= \mathbf{w}^{t-1} - \alpha \mathbf{L}_\mathbf{w} \\
&= \begin{bmatrix} 2 & 1 & 3 \end{bmatrix} - 0.01 \begin{bmatrix} 44 & 72 & 100 \end{bmatrix} \\
&= \begin{bmatrix} 1.56 & 0.28 & 2 \end{bmatrix} \\
\mathbf{b}^t &= \mathbf{b}^{t-1} - \alpha \mathbf{L}_\mathbf{b} \\
&= 1 - 0.01 \cdot 28 \\
&= 0.72 \\
\hat{y}_1^t &= x_1 w_1^t + x_2 w_2^t + x_3 w_3^t + b^t \\
&= 8.84 \\
\hat{y}_2 &= x_2 w_1^t + x_3 w_2^t + x_4 w_3^t + b^t \\
&= 12.68 \\
\hat{\mathbf{y}}^t &= \begin{bmatrix} 8.84 & 12.68 \end{bmatrix} \\
\mathcal{L}(\hat{y}, y)^t &= \frac{1}{2}[(8.84 - 2)^2 + (12.68 - 4)^2] \\
&\sim 61.06
\end{aligned}$$

Loss has decreased.

Equivariance: Questions

I will address both convolution and cross-correlation as convolution in this section for convenience.

1)

Network	Original data	Shifted data
MLP	0.9432	0.4147
Convolutional model	0.9612	0.5556

Table 1: Accuracies of the resulting models in the experiment

The results for CNNs are explained in question 3 of this section. Matrix multiplications are not equivariant at all. Adding into account the data loss by shifting, MLPs should perform poorly on shifted data as well.

However, convolution layers on bounded data are *partially* equivariant (made up term). Depending on the number of pixels shifted, if this number is small, then the output will likely be more similar compared to the output of an input with larger number of shifts. Choosing sufficiently small number (i.e. `shift = 2` in the assignment text), output for the CNN will likely be more similar than that of MLPs', hence the CNNs should perform better compared to MLPs, as seen in table 1.

2)

Data augmentation (especially introducing translational variants).

3)

The convolution operations are equivariant when the domain of the function (i.e. width and height for 2D data) is infinitely large. However, in a practical application, where the data comes with finite dimensions, the part of the data may be lost due to shifting operations. Hence the output of a bounded data is a cropped version of the output of an unbounded theoretical data (different even when normalized). The accuracies may be very different for these outputs. Most likely, the output coming from input with more data will have a larger accuracy score, as seen in table 1.