Exercise 2 - Convolutions and Loss Functions

Abdalla Arafa, 428044
Ehsan Attar, 427214
Sai Leela Poduru, 428272
Prateek Rathod, 428396
Shruti Shrivastava, 428364
Erik Schwede, 428240

November 2023

2.1. Convolutions

i)

-8	-5	-10	11	18
-19	-5	-2	0	21
-23	4	-3	-14	24
-12	7	-17	-9	29
-3	1	-19	6	22

-10	-17	-18	-17	-16
-1	-5	4	0	-6
-9	6	5	-3	-6
8	13	3	5	13
13	11	13	23	22

33	13	-20	-18	3
5	4	-6	22	-16
-2	-23	15	-16	-4
-13	15	4	4	-18
-13	8	-12	-19	12

ii)

These filters are commonly used for edge detection in image processing:

- Prewitt Filter: Emphasizes both horizontal and vertical edges in an image.
- Laplacian Filter: Emphasizes regions of rapid intensity change.
- Sobel Filter: Emphasizes vertical edges in an image.

iii)

In CNNs, ”valid” padding involves no additional padding, resulting in a smaller output, while ”same” padding maintains input dimensions by adding padding as needed, yielding an output size equal to the input.

iv)

CNNs are preferred over MLPs for image data due to their ability to capture spatial hierarchies efficiently. The use of convolutional layers allows them to recognize local patterns and complex visual features, while weight sharing provides translation invariance. CNNs naturally form feature hierarchies, making them adept at image recognition tasks by exploiting spatial correlations. These architectural features make CNNs more effective for handling the inherent characteristics of image data compared to traditional MLPs.

v)

$$O = \frac{H - k + 2p}{s} + 1$$

vi)

$$\text{Trainable parameters} = N \times (k \times k \times C + 1)$$

where: N is the number of filters, k is the size of each filter (assuming square filters, $k \times k$), C is the number of channels in the input, and 1 accounts for the bias term associated with each filter.

2.2 Loss Functions and Optimization

i) Derivative of softmax

$$\hat{y_i} = \frac{e^{z_i}}{\sum_{k=1}^{n} e^{z_k}} \tag{1}$$

$$\tag{2}$$

Derivative for condition i = j using quotient rule:

$$\begin{aligned} \frac{\partial \hat{y_i}}{\partial z_i} &= \frac{e^{z_i}\sum e^{z_k} - e^{z_i}.e^{z_i}}{(\sum_k e^{z_k})^2} \qquad (3) \\ &= \frac{e^{z_i}(1 - e^{z_i})}{(\sum_k e^{z_k})^2} \qquad (4) \\ &= \hat{y_i}(1 - \hat{y_i}) \qquad (5) \end{aligned}$$

Derivative for condition i != j using quotient rule:

$$\begin{aligned} \frac{\partial \hat{y_i}}{\partial z_j} &= \frac{0 - e^{z_i}.e^{z_j}}{(\sum_k e^{z_k})^2} \qquad (6) \\ &= -\hat{y_i}.\hat{y_j} \qquad (7) \end{aligned}$$

ii) Derivative of cross-entropy loss function defined as

$$L(y, \hat{y}) = -\sum_{k=1}^{N} y_k \log \hat{y} \tag{8}$$

$$\frac{\partial L}{\partial \hat{y_i}} = -\sum_{i} y_i.\frac{1}{\hat{y_i}} \tag{9}$$

Using chain rule:

$$\frac{\partial L}{\partial z_i} = \frac{\partial L}{\partial \hat{y_i}}.\frac{\partial \hat{y_i}}{\partial z_i} \tag{10}$$

$$\begin{aligned} \frac{\partial L}{\partial z_i} &= -\sum_{i \neq j} y_i.\frac{1}{\hat{y_i}}\frac{\partial \hat{y_i}}{\partial z_i} - y_i.\frac{1}{\hat{y_i}}\frac{\partial \hat{y_i}}{\partial z_i} \qquad (11) \\ &= -\sum_{i \neq j} y_i.\frac{1}{\hat{y_i}}.(-\hat{y_i}\hat{y_j}) - y_i.\frac{1}{\hat{y_i}}.\hat{y_i}(1 - \hat{y_i}) \qquad (12) \\ &= \sum_{i \neq j} y_i.\hat{y_j} + y_i\hat{y_i} - y_i \qquad (13) \\ &= \sum_{i} y_i.\hat{y_j} - y_i \qquad (14) \\ &= \hat{y_j} - y_i \qquad (15) \end{aligned}$$

2.3 Loss Functions with Regularization

Given,

$$\hat{y} = Xw$$

$$\text{Ł}(y, \hat{y_i}) = \frac{1}{n} * \sum_{i=1}^{n}((\hat{y_i} - y_i)^2) + \sum_{i=1}^{d} w_i^2$$

$$= \frac{1}{n} * \sum_{i=1}^{n}((\hat{y_i} - y_i)^2) + ||w||_2^2$$

Now, Let's find derivative of L w.r.t w and set it to 0 to get minimum value of w

$$\frac{\partial L}{\partial w} = 0$$

$$\implies \frac{1}{n} * \sum_{i=1}^{n} 2(\hat{y_i} - y_i)(X_i) + 2\lambda w = 0$$

$$\implies \frac{1}{n} * X^T(\hat{y} - y) + \lambda w = 0$$

$$\implies w = \frac{1}{\lambda n} * X^T(y - \hat{y})$$

$$\therefore Closed\ form\ for\ vector\ w\ that\ minimizes\ L\ is$$

$$w = \frac{1}{\lambda n} * X^T(y - Xw)$$