

Exercise 2 - Convolutions and Loss Functions

Deadline: 28.11.2022

Total Marks: 30

Submission

- Submissions through OLAT. Only one group member needs to submit it.
- Your submission should contain a PDF with the solutions to the exercise questions (and any Python code files) zipped together in a single file.
- Include the group number along with the names and matriculation numbers of all group members on the PDF.
- For Jupyter notebooks, please save them with the outputs of your code displayed.
- Question 2.6 is not mandatory. Bonus points can be used to make up for lost points in other exercises as well.

2.1. Convolutions

[3+1+1+1+1+1=8]

Given the following 5x5 input image with one channel:

5	5	2	5	5
5	5	2	5	5
7	7	5	7	7
5	5	2	5	5
5	5	2	5	5

Let's assume we have the following 3x3 filters:

$$\begin{vmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{vmatrix} \quad \begin{vmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{vmatrix} \quad \begin{vmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{vmatrix}$$

Tasks:

i) Apply the given filters (by cross-correlation) to the above dataset, i.e.:

$$Y_{i,j} = (K \star X)_{i,j} = \sum_{m=0}^2 \sum_{n=0}^2 K_{m,n} X_{i+m,j+n} \quad (1)$$

where Y is the output. Use 'same' padding and a stride of one.

ii) Look at the structure of the filters. What do they do?

iii) What is the difference between 'valid' and 'same' padding?

iv) Why do we prefer Convolutional Neural Networks (CNNs) over a Multi-Layer Perceptron (MLP) for image data?

v) Given an input image of size $H \times H$, a convolutional filter of size k , padding p and stride s . Write down the formula for calculating the dimensions of the output of the convolution operation.

vi) Given an input of size (H, W, C) convolved with N Conv2D filters of size k , what are the number of trainable parameters in this convolutional layer? Write down the formula. Assume both weight and bias are present.

2.2. Loss Functions and Optimization [1+1+1+2+2=7]

i) What is a loss function in deep learning? How is it different from an evaluation metric?

ii) What is the main difference between regression and classification tasks? Name one commonly used loss function for each of these.

iii) If during training, the validation loss is not decreasing but the training loss is, what does this indicate?

iv) Given $\hat{y} = \text{softmax}(z)$ with

$$\hat{y}_i = \frac{e^{z_i}}{\sum_{k=1}^N e^{z_k}},$$

where $\hat{y} \in \mathbb{R}^N$ and N is the number of classes of a classification problem. Calculate $\frac{\partial \hat{y}_i}{\partial z_j}$.

v) Given $\hat{y} = \text{softmax}(z)$, a target vector $y \in \mathbb{R}^N$ and the cross-entropy loss function defined as

$$L(y, \hat{y}) = - \sum_{k=1}^N y_k \log \hat{y}_k. \quad (2)$$

Calculate $\frac{\partial L}{\partial z_i}$. *Hint: Make use of the chain rule $\frac{\partial L}{\partial z_j} = \sum_i \left(\frac{\partial L}{\partial \hat{y}_i} \frac{\partial \hat{y}_i}{\partial z_j} + \frac{\partial L}{\partial y_i} \frac{\partial y_i}{\partial z_j} \right)$. Moreover, reuse the results of 2.2(iv) and the fact that vector y contains the probabilities for each class i that sum up to one, i.e: $\sum_i y_i = 1$.*

2.3. Mean Squared Error [5]

Consider the input dataset $X \in \mathbb{R}^{n \times d}$ with n samples of size d , a target vector $y \in \mathbb{R}^n$, a weight vector $w \in \mathbb{R}^d$ and a prediction $\hat{y} = Xw$. The mean squared error (MSE) is defined as the sum of the squared differences between the prediction \hat{y}_i and the true values y_i for each instance:

$$L(y, \hat{y}) = \frac{1}{n} \sum_{i=1}^n (\hat{y}_i - y_i)^2, \quad (3)$$

where $\hat{y}_i = w^T x_i$ and $x_i \in \mathbb{R}^d$ is one sample of dataset (corresponding to a row in X).

Task: Find the vector w that minimizes the MSE loss function!

Hint: You can write the sum above as a vector product! Moreover, you can use the following identities: $(Ax)^T = x^T A^T$, $\frac{\partial x^T A x}{\partial x} = 2Ax$ and $\frac{\partial x^T b}{\partial x} = b$ for vectors x and b and matrix A .

2.4. CIFAR Challenge [5]

Follow the instructions in the Jupyter notebook `Task_2.4.ipynb` to complete the CIFAR competition using PyTorch. Your task is to fill in the missing code annotated with `TODO` tags in the comments, and get an accuracy of at least 70%.

2.5. Depthwise Separable Convolutions [5]

In this task, you will explore depth-wise separable convolutions, which is a special type of convolution. Follow the instructions in the Jupyter notebook `Task_2.5.ipynb` and fill in the missing code annotated with `TODO` tags.

2.6. Dog Breeds Identification on Kaggle [Bonus]

See the notebook `Task_2.6_DogBreedsIdentification.ipynb` for details of the task. This exercise is optional. You can get up to 5 bonus points here. Good luck!