Aufgabe 1: Convolutions

You are given an input feature map with **2 channels of size** 5×5:

$$X = \begin{bmatrix} \text{Channel 1} \\ \text{Channel 2} \end{bmatrix}$$

Channel 1:

$$\begin{bmatrix} 1 & 0 & 2 & 1 & 3 \\ 4 & 1 & 0 & 2 & 1 \\ 1 & 3 & 2 & 0 & 2 \\ 0 & 1 & 3 & 1 & 1 \\ 2 & 2 & 1 & 0 & 0 \end{bmatrix}$$

Channel 2:

$$\begin{bmatrix} 0 & 1 & 2 & 1 & 0 \\ 1 & 0 & 1 & 2 & 2 \\ 2 & 1 & 0 & 1 & 1 \\ 1 & 2 & 1 & 0 & 0 \\ 0 & 1 & 2 & 1 & 1 \end{bmatrix}$$

First, apply a **standard 2D convolution** with the following parameters:

- Input channels: 2
- Output channels: 1
- Kernel size: 2×2
- Stride: 2
- Padding: 0

The convolution uses the following kernel:

Channel 1 kernel: $\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$ Channel 2 kernel: $\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$

1.1 Calculate output shape

First, calculate the output shape after this convolution. Write down the formula with which you calculated it.

1.2 Calculate output feature map

Compute the output feature map of this convolution by hand.

1.3 Depthwise Separable Convolution

Next, consider how this convolution can be rewritten as a **depthwise separable convolution**. Describe the two steps required and specify the parameters (number of filters, kernel size, stride, padding) of both convolutions. Explain how they relate to the original convolution.

1.4 What's a bottleneck layer and why is it used?

1.5 Describe the advantages and drawbacks of grouped convolutions. How can the drawbacks be alleviated?

Aufgabe 2: Designing Efficient CNNs for FashionMNIST

In this exercise we want to design and implement our own efficient CNN in pytorch with the techniques introduced in the lecture. Because we cannot calculate the latency on the hardware yet, we use the amount of FLOPs as a proxy metric for now. Keep in mind that FLOPs only correlate losely with latency.

2.1 Baseline Model

Before designing an efficient architecture we want to implement a baseline model to compare to. For that design a CNN that achieves at least a **Test Accuracy $\geq$ 91%** on FashionMNIST. What is the architecture? How did you train it?

Tip: Pytorch Layers to consider

- `nn.Conv2d`
- `nn.BatchNorm2d`
- `nn.MaxPool2d`
- `nn.Dropout`
- `nn.Linear`
- Activation functions (e.g., ReLU)

2.2 Resource Analysis

Analyse your baseline model according to its resources used. What are the:

- Total parameters
- FLOPs per forward pass for one sample

For the number of parameters you may use the function implemented in the previous exercise. For the FLOPs (Floating Point Operations) a function has to be implemented. For this, you may use the `torch.utils.flop_counter` package or the formulas given in the lecture.

2.3 Model Optimization

Design a more efficient CNN with the techniques introduced in the lecture. Your model should fulfill at least the following constraints:

- Test accuracy $\geq$ 90%
- FLOPs $<$ 500,000
- Parameters $<$ 80,000

Describe the final model architecture, FLOP and parameter count and accuracy. Which techniques had which impact? What worked better or worse.

Techniques to consider:

- Depthwise separable convolutions
- Bottleneck layers
- Grouped convolutions
- Channel shuffling