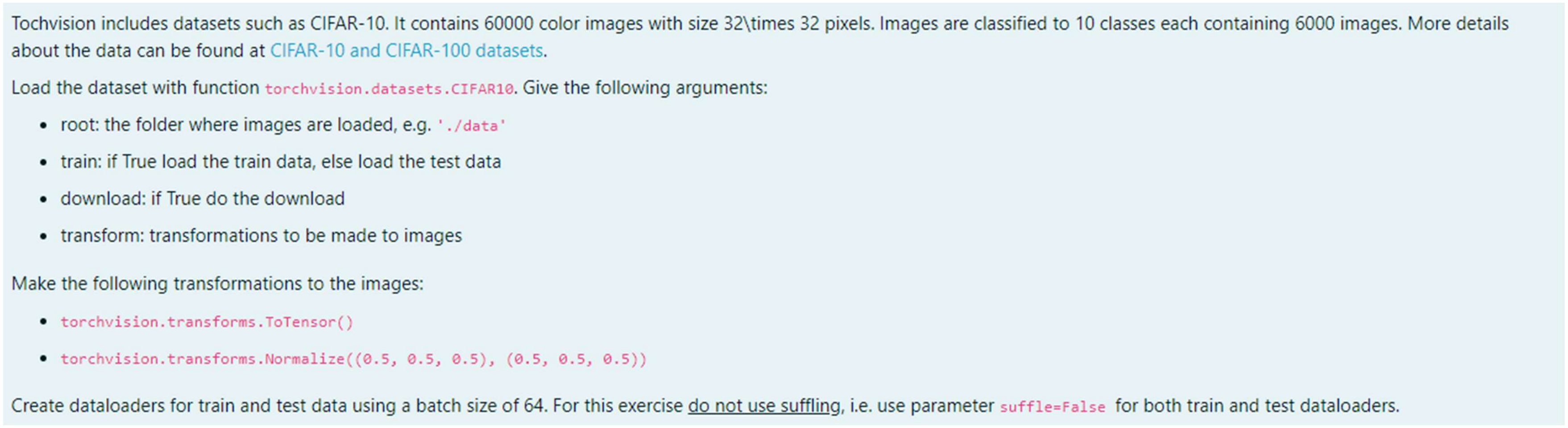
Exercise 11.1



Tochvision includes datasets such as CIFAR-10. It contains 60000 color images with size 32\times 32 pixels. Images are classified to 10 classes each containing 6000 images. More details about the data can be found at CIFAR-10 and CIFAR-100 datasets.

Load the dataset with function `torchvision.datasets.CIFAR10`. Give the following arguments:

- root: the folder where images are loaded, e.g. `'./data'`
- train: if True load the train data, else load the test data
- download: if True do the download
- transform: transformations to be made to images

Make the following transformations to the images:

- `torchvision.transforms.ToTensor()`
- `torchvision.transforms.Normalize((0.5, 0.5, 0.5), (0.5, 0.5, 0.5))`

Create dataloaders for train and test data using a batch size of 64. For this exercise do not use suffling, i.e. use parameter `suffle=False` for both train and test dataloaders.

Create convolution neural network that has the following structure:

- input is a image of shape 32×32 with 3 input channels
- convolution part
    - convolution layer (Conv2d) where input channels is 3 (color image) , number of filters (out_channels) is 16, filter size is 3 (i.e. fiter is of shape 3×3), striding is 1and padding is 1
        - this outputs a shape of $32 \times 32 \times 16$
    - activation ReLU
    - pooling layer (MaxPool2d): MaxPooling with filter size 2 (i.e. filter is of shape 2×2) and striding 2.
        - this outputs a shape of $16 \times 16 \times 16$
    - convolution layer (Conv2d) where input shape is 16 (based on previous layer's output), number of filters (out channels) is 32, filter size is 3, striding is 1 and padding is 1
        - this outputs a shape of $16 \times 16 \times 32$
    - activation ReLU
    - pooling layer (MaxPool2d): MaxPooling with filter size 2 (i.e. filter is of shape 2×2) and striding 2.
        - this outputs a shape of $8 \times 8 \times 32$
- classifier part
    - previous layer's output as a vector
    - hidden layer of 128 neurons
    - activation ReLU
    - output layer with 10 neurons
    - activation Softmax

Note that the activation Softmax can be done after neural network has outputted, i.e. you don't have to include it to the CNN.

Use Adam as a optimizer with learning rate 0.001 and cross entropy as loss function.

Use the seed to 121 just before creating the model (`torch.manual_seed`). This is crucial so that you can get to the right answer to the last question in the exercise.

What is the number of parameters in CNN created? (Tip: Use the `torchinfo.summary` function with parameter `input_size=[64, 3, 32, 32]`).

Make 10 epochs and get the train loss and train accuracy and the test accuracy.

What is the test accuracy after 10th epoch? Give the answer as an decimal number rounded to two decimals, e.g. an accuracy of 45.55% is given in the form of 0.46.