Assignment 3

Due week 7 before class time

Given a 32x32 pixels, 3 channels input. Fill the pixel values with torch.randn(. . .)

Foreach pytorch functions in the list,

1. Initialise the weights with uniform random numbers r
2. Call the functions and get the output tensors - torch_out
3. Implement these functions from scratch, without using any neural network libraries. Use linear algebra libraries in python is ok. Output your tensors as — my_out
4. Compare and show that torch_out and my_out are equal up to small numerical errors

1. `torch.nn.MaxPool2d(kernel_size=2, stride=1, padding=0, dilation=1, return_indices=False, ceil_mode=False)`

2. `torch.nn.AvgPool2d(kernel_size=2, stride=1, padding=0, ceil_mode=False, count_include_pad=True, divisor_override=None)`

3. `torch.nn.Conv2d(in_channels=3, out_channels=6, kernel_size=3, stride=1, padding=0, dilation=1, groups=1, bias=True, padding_mode=‘zeros')`

4. `torch.nn.Conv2d(in_channels=3, out_channels=6, kernel_size=5, stride=2, padding=0, dilation=2, groups=1, bias=True, padding_mode=‘zeros')`

5. `torch.nn.ConvTranspose2d(in_channels=3, out_channels=4, kernel_size=3, stride=1, padding=0, output_padding=0, groups=1, bias=True, dilation=1, padding_mode='zeros')`

1. `torch.flatten(input, start_dim=0, end_dim=-1)`

2. `torch.sigmoid(input, *, out=None)`

3. `torchvision.ops.roi_pool(input: torch.Tensor, boxes: torch.Tensor, output_size: None, spatial_scale: float = 1.0)`

4. `torch.nn.functional.batch_norm(input, running_mean, running_var, weight=None, bias=None, training=False, momentum=0.1, eps=1e-05)`

5. `torch.nn.functional.cross_entropy(input, target, weight=None, size_average=None, ignore_index=-100, reduce=None, reduction=‘mean')`

6. `torch.nn.functional.mse_loss(input, target, size_average=None, reduce=None, reduction='mean')`