

NEURAL NETWORKS – exercise 6

In exercise 6, you should build a network using CNN layers and the max pooling operator. The CNN layer is available at `torch.nn.Conv2d`:

<https://pytorch.org/docs/stable/generated/torch.nn.Conv2d.html>

The layer accepts 4-dimensional inputs (batch, image channels, height, width). For black and white images, the number of input channels will be equal to 1, the number of output channels should be greater. After passing through one or more convolutional layers, the output must be flattened and fed to the linear layer, and then use the standard cost function for classification.

To implement a linear layer, you can use `LazyLinear`, which automatically establishes dimensionality on the first pass of the data.

It will need to be examined

- The number of output channels of the convolutional layer
- Convolutional layer filter size
- Pooling window size
- Data disturbance: data can be distorted by adding to the input batch a batch of the same dimensions, generated as Gaussian noise with different deviations. Explore scenarios: noise added in test data vs. noise added in both test and training data.

The exercise is assessed on a scale of 0-10 points and you have 2 weeks to complete it.