

# CNN Lab

## Ex. 1

Implement a LeNet-like architecture to classify handwritten digits. Compared to the original architecture, we will use 28x28 (instead of 32x32) input images and standard linear layers (we will not consider Gaussian connections).

```
class LeNet(nn.Module):
    def __init__(self, output_dim):
        super().__init__()

        self.features = nn.Sequential(
            nn.Conv2d(in_channels=1, out_channels=6, kernel_size=5, padding=2),
            nn.MaxPool2d(kernel_size=2),
            nn.ReLU(),
            nn.Conv2d(in_channels=6, out_channels=16, kernel_size=5),
            nn.MaxPool2d(kernel_size=2),
            nn.ReLU()
        )

        self.linear = nn.Sequential(
            nn.Linear(16 * 5 * 5, 120),
            nn.ReLU(),
            nn.Linear(120, 84),
            nn.ReLU(),
            nn.Linear(84, output_dim)
        )

    def forward(self, x):
        x = self.features(x)
        x = x.view(x.shape[0], -1)
        x = self.linear(x)

        return x
```

## Ex. 1.1

Compute the number of parameters for each layer of the previous network.

Number of parameters (Conv Layer) = Filter Size x Filter Size x Number of Filters x Input Depth + Biases

Number of parameters (Linear Layer) = Input Size x Output Size + Biases

* Layer C1	-> 5x5x6x1+6	= 156
* Layer C3	-> 5x5x16x6+16	= 2,416
* Layer C5	-> 5x5x16x120+120	= 48,120
* Layer F6	-> 120*84+84	= 10,164
* Layer OUT	-> 84*10+10	= 850

---

Total number of parameters	= 61,706
----------------------------	----------