

## Звіт

1. Було побудовано модель з наступною архітектурою та вираховано кількість вагів і форму тензорів на кожному шарі:

```
class Net(nn.Module):
    def __init__(self):
        super(Net, self).__init__()
        self.conv1 = nn.Conv2d(1, 16, 9, 1)      # <- (9*9 + 1)*16 = 1 312
        self.conv2 = nn.Conv2d(16, 4, 9, 1)      # <- (9*9)*16*4 + 4 = 5 188
        self.dropout1 = nn.Dropout(0.25)        # <- 0
        self.dropout2 = nn.Dropout(0.5)         # <- 0
        self.fc1 = nn.Linear(144, 50)           # <- 144*50 + 50 = 7 250
        self.fc2 = nn.Linear(50, 10)            # <- 50*10 + 10 = 510

    def forward(self, x):
        # [128, 1, 28, 28]
        x = self.conv1(x)                       # [128, 16, 20, 20]
        x = F.relu(x)                           # [128, 16, 20, 20]
        x = self.conv2(x)                       # [128, 4, 12, 12]
        x = F.relu(x)                           # [128, 4, 12, 12]
        x = F.max_pool2d(x, 2)                  # [128, 4, 6, 6]
        x = self.dropout1(x)                    # [128, 4, 6, 6]
        x = torch.flatten(x, 1)                 # [128, 4*6*6 = 144]
        x = self.fc1(x)                         # [128, 50]
        x = F.relu(x)                           # [128, 50]
        x = self.dropout2(x)                    # [128, 50]
        output = self.fc2(x)                    # [128, 10]
        return output
```

```
Test set: Average loss: 0.0020, Accuracy: 9259/10000 (93%)
# Params: 14260
Net(
  (conv1): Conv2d(1, 16, kernel_size=(9, 9), stride=(1, 1))
  (conv2): Conv2d(16, 4, kernel_size=(9, 9), stride=(1, 1))
  (dropout1): Dropout(p=0.25, inplace=False)
  (dropout2): Dropout(p=0.5, inplace=False)
  (fc1): Linear(in_features=144, out_features=50, bias=True)
  (fc2): Linear(in_features=50, out_features=10, bias=True)
)

=====
Layer (type)          Output Shape          Param #
=====
Conv2d-1              [-1, 16, 20, 20]      1,312
Conv2d-2              [-1, 4, 12, 12]       5,188
Dropout-3             [-1, 4, 6, 6]         0
Linear-4              [-1, 50]              7,250
Dropout-5             [-1, 50]              0
Linear-6              [-1, 10]              510
=====
Total params: 14,260
```

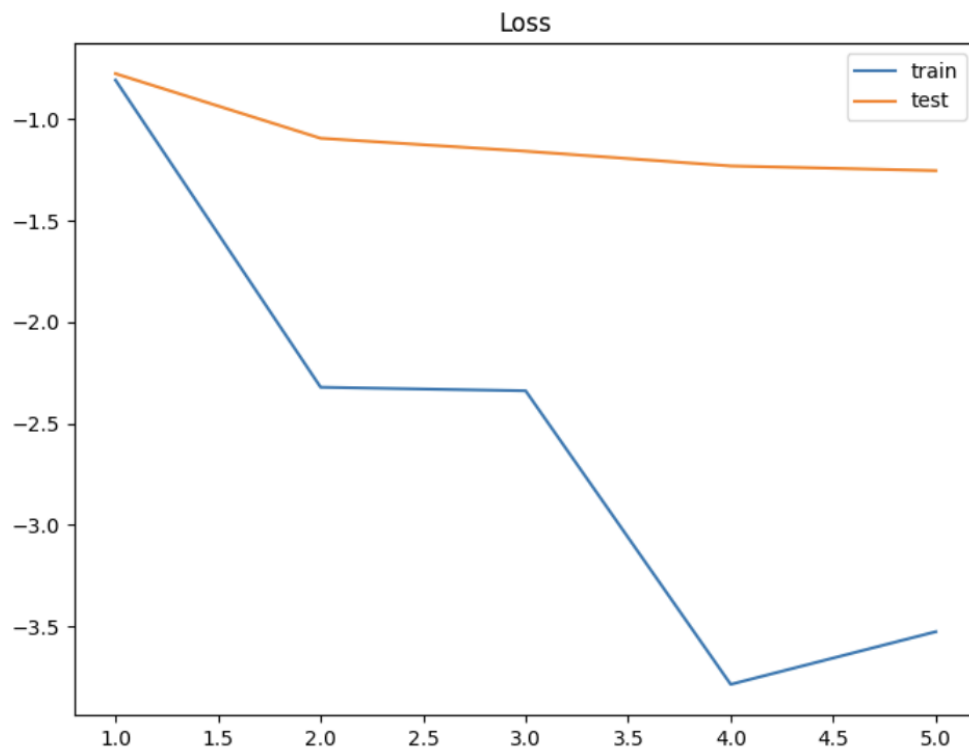
При зменшенні кількості параметрів до 14 260 точність прогнозів знизилася на 3% і становить 93% порівнюючи з оригінальною моделлю (1 199 882 параметрів, 96% точності).

2. Було побудовано модель з overfit, яка мала велику кількість параметрів (483 482) та навчалась на маленькому датасеті (1 200 картинок).

```
# Params: 483482
Net(
  (fc1): Linear(in_features=784, out_features=528, bias=True)
  (fc2): Linear(in_features=528, out_features=128, bias=True)
  (fc3): Linear(in_features=128, out_features=10, bias=True)
)
```

| Layer (type) | Output Shape | Param # |
|--------------|--------------|---------|
| Linear-1     | [-1, 528]    | 414,480 |
| Linear-2     | [-1, 128]    | 67,712  |
| Linear-3     | [-1, 10]     | 1,290   |

Total params: 483,482



З графіка функції Loss бачимо, що модель має значно кращі результати на train, ніж на test, що свідчить про overfit.