

Vaja 4

Podatki:

Širina slik: 28

Višina slik: 28

Število kanalov: 1

Število slik: 60000

```
class ImageDataSet(Dataset):
    def __init__(self):
        mnist_ds = torchvision.datasets.MNIST(
            root="datasets", train=True, transform=torchvision.transforms.ToTensor(),
            download=True)
        print(f"Širina slik: {mnist_ds[0][0].shape[2]}")
        print(f"Višina slik: {mnist_ds[0][0].shape[1]}")
        print(f"Število kanalov: {mnist_ds[0][0].shape[0]}")
        print(f"Število slik: {len(mnist_ds)}")
        print(f"Podatkovna zbirka: MNIST (http://yann.lecun.com/exdb/mnist/)")

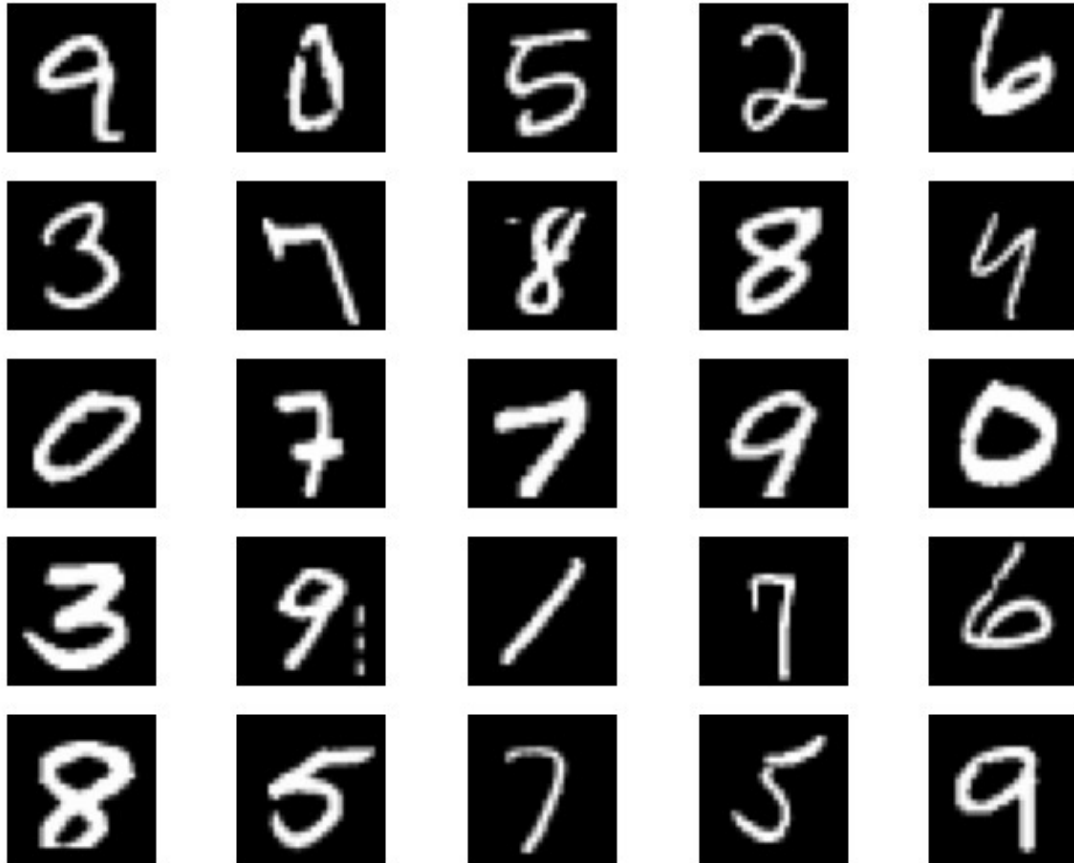
        n_rows = 5
        n_cols = 5
        _, axes = plt.subplots(n_rows, n_cols)
        for r in range(n_rows):
            for c in range(n_cols):
                img, _ = mnist_ds[random.randint(0, len(mnist_ds) - 1)]
                axes[r, c].imshow(img.permute(1, 2, 0), cmap="gray")
                axes[r, c].axis("off")

        plt.tight_layout()
        plt.show()

        self.images = mnist_ds

    def __len__(self):
        return len(self.images)

    def __getitem__(self, index):
        img, _ = self.images[index]
        img = img/255
        return img
```



Nevronska mreža:

Diskriminator

=====		
Layer (type:depth-idx)	Output Shape	Param #
=====		
Discriminator	[256, 1]	--
└─Conv2d: 1-1	[256, 64, 14, 14]	640
└─BatchNorm2d: 1-2	[256, 64, 14, 14]	128
└─LeakyReLU: 1-3	[256, 64, 14, 14]	--
└─Conv2d: 1-4	[256, 64, 7, 7]	36,928

└─BatchNorm2d: 1-5	[256, 64, 7, 7]	(recursive)
└─LeakyReLU: 1-6	[256, 64, 7, 7]	--
└─Flatten: 1-7	[256, 3136]	--
└─Linear: 1-8	[256, 1]	3,137
└─Sigmoid: 1-9	[256, 1]	--

=====

Total params: 40,833

Trainable params: 40,833

Non-trainable params: 0

Total mult-adds (Units.MEGABYTES): 496.21

=====

Input size (MB): 0.80

Forward/backward pass size (MB): 64.23

Params size (MB): 0.16

Estimated Total Size (MB): 65.19

=====

Generator

=====

Layer (type:depth-idx)	Output Shape	Param #
=====		
Generator	[256, 1, 28, 28]	--
└─Linear: 1-1	[256, 6272]	633,472
└─ReLU: 1-2	[256, 6272]	--
└─ConvTranspose2d: 1-3	[256, 128, 14, 14]	262,272

—BatchNorm2d: 1-4	[256, 128, 14, 14]	256
—ReLU: 1-5	[256, 128, 14, 14]	--
—ConvTranspose2d: 1-6	[256, 128, 28, 28]	(recursive)
—BatchNorm2d: 1-7	[256, 128, 28, 28]	(recursive)
—ReLU: 1-8	[256, 128, 28, 28]	--
—Conv2d: 1-9	[256, 1, 28, 28]	6,273
—Sigmoid: 1-10	[256, 1, 28, 28]	--

=====

Total params: 902,273

Trainable params: 902,273

Non-trainable params: 0

Total mult-adds (Units.GIGABYTES): 67.22

=====

Input size (MB): 0.10

Forward/backward pass size (MB): 528.25

Params size (MB): 3.61

Estimated Total Size (MB): 531.96

=====

Parametri učenja:

```

disc_model = Discriminator()
disc_optim = torch.optim.Adam(disc_model.parameters(), lr=0.0002, betas=(0.5, 0.999))
disc_loss_fn = nn.BCELoss()

gen_model = Generator()
gen_optim = torch.optim.Adam(gen_model.parameters(), lr=0.5, betas=(0.5, 0.999))
gen_loss_fn = nn.BCELoss()

data_set = ImageDataSet()
data_gen = DataLoader(data_set, batch_size=128, shuffle=True)

train(disc_model, disc_optim, disc_loss_fn, gen_model, gen_optim, gen_loss_fn,
      100, 128, data_gen, create_identity(1))

```

Rezultati:

