

Exercise 4

Due on: Thursday, 23.05.2024

Task 11 PyTorch Continued

The last task was about implementing a 10-class classifier for the MNIST data set using PyTorch.

- (i) Add a dropout layer* between the fully connected linear layers of the classifier. Test several values of the dropout probability p and report on the train and test accuracy. Use a learning rate of 0.01 and train for at least 15 epochs. Which p works best?
- (ii) Visualize the learned filters of the convolutional layers. You can access the weights via `net.conv1.weight.data.cpu().numpy()`
- (iii) Take a training sample and manually apply every operation of the forward pass. Take a look at the intermediate results.

Task 12 Logistic Regression using PyTorch

This task is about implementing a logistic regression (binary classifier) using PyTorch.

- (i) Use the MNIST data set and consider the binary classification task of 1 vs. 7.
- (ii) Define an appropriate model structure and loss function.
- (iii) Train your classifier and report on the train and test accuracy.
- (iv) Reshape the learned weight matrix such that it has the same dimensionality as the input and visualize it.
- (v) Add an ℓ_1 and ℓ_2 regularizer. Conduct some experiments by changing λ (in both cases for ℓ_1 and ℓ_2) and report on the train and test accuracy.

*<https://pytorch.org/docs/stable/generated/torch.nn.Dropout.html#torch.nn.Dropout>