NEURAL NETWORKS – exercise 5

In exercise 5, we will use a multi-layer network to classify images. We will use the FashionMNIST collection, which is available by default in torchvision:

```
data = torchvision.datasets.FashionMNIST('path',download =True)
data_loader = torch.utils.data.DataLoader(data,
                                          batch_size=16,
                                          shuffle=True)
```

DataLoader is an iterator that returns batched input-output pairs, the output in this case is one of ten classes denoting the types of clothes we see in the image.

Due to the fact that the data are images, they will need to be transformed into a feature matrix that the linear layer can accommodate, flattening the tensor (methods operating on the shape of the tensor are available in torch.tensor, there is also a Flatten layer).

The results and learning curves should be compared for one- and two-layer neural networks depending on:

- Number of neurons in the hidden layer
- Batch size
- Number of training examples (use subsets of the training set of different sizes instead of the full training set - check the sizes of 1% of the data, 10% of the data)
- 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 graded on a scale of 0-10 points.