Exercise (5th Day)
(Neural Networks)

In this task you will familiarize yourself with how neural networks can be used for different classification tasks. First you will design a single layer network to solve a simple two class problem on a set of two dimensional datapoints. In the second task an end-to-end CNN architecture is implemented to classify two classes of MNIST digits using PyTorch.

**Exercise 1:** Classify the dataset data2d with a single layer neural network. First, consider which sort of classification problems can be solved by this simple network topology. Choose the training and test datasets accordingly.

1. Implement a neural network with two input- and a single output neuron. As a starting point, the output neuron should have a linear activation function. Do not neglect a bias! Visualize your results for different training iterations. **Tips:** Such a neural network can be evaluated by a single matrix multiplication. Average the update of the network weights over all training samples. Choose a small learning rate.
2. What accuracy do you achieve on the test set?
3. What is the purpose of a bias?
4. What influence do different learning rates have?
5. How does different activation functions affect classification accuracy?
6. (optional) Extent the neural network such that it is able to solve any arbitrary two class problem.
7. (optional) Extent the neural network such that it can solve a three class problem.

**Exercise 2:** Suppose we have a dataset consisting of two classes of MNIST digits and we want to separate them based on their classes. Design the following architecture\[1\] using PyTorch to solve the question.

In general a simple CNN for MNIST classification could have the architecture [INPUT - CONV - RELU - FC]. In more detail:

INPUT [64x64x1] will hold the raw pixel values of the image, in this case an image of width 64, height 64, and with one color gray scale channel.

CONV layer will compute the output of neurons that are connected to local regions in the input. This may result in volume such as [64x64x10] if we decided to use 10 filters.

RELU layer will apply an elementwise activation function. FC (i.e. fully-connected) layer will compute the class scores, resulting in volume of size [1x1x2], where each of the 2 numbers correspond to a class score, such as among the 2 categories of our subset of MNIST.

Optional: Run the model with the whole MNIST dataset and change the architecture to classify the dataset to improve and get the best result in the test dataset. For solving this task use the **Pytorch documentation**

Abbildung 1: Convolutional Neural Network Architecture