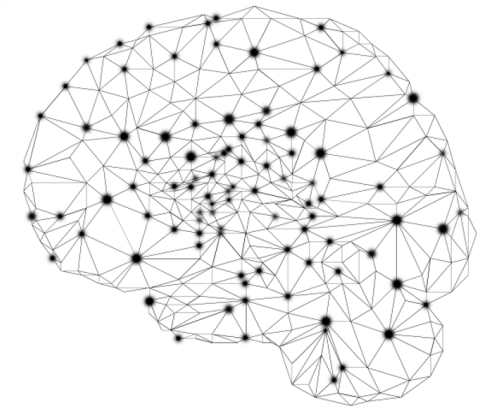

Neural Networks and Data Science

Lab #6

15.11.2023
Deadline: 22.11.2023, 12:10

Dr. Marcel Völschow

HAW HAMBURG

Problem 1

Neural networks need tons of labelled data. The `MNIST` dataset of labelled handwritten digits is one of the basic benchmarks to check the predictive power of a network. Here, we will load some of the data into a Numpy array for further processing.

a) `mnist_test.zip` contains 10,000 images of handwritten digits following the name scheme

`test_XXXXX.gif`

where `XXXXX` is an integer with a field length of 5 and padding zeros. In addition, the archive contains a text file called `labels.txt` that holds the labels of the images, where line 1 corresponds to the first image, line 2 to the second image, and so on. Create a folder `data` and unzip the contents of the archive using

`unzip -q mnist_test.zip -d data/`

b) Create a 1D Numpy array labels and read in all labels from `labels.txt`.

c) Create a 3D Numpy array test. The first dimension shall represent the image index (0 to 9999) while the second and third dimension represent the image pixels.

d) Generate a list of all images imgNames using list comprehension. Hint: If your images are within a folder `test/` (relative to the Notebook), you can use

```
filename="test/test_"+str(i).zfill(len(str(n)))+".png"
```

where i is the image index and `n` is the total number of images. You can also try `listdir` from the `os` module.

e) Iterate through all image files. In every iteration, load the associated image file, convert it to a Numpy array, divide it by 255 to normalize it to 1 and store it in test.

Problem 2

Thanks to `Pillow`, a few lines of code suffice to get handwritten stuff into your notebook.

a) Open `Microsoft Paint`, click on the `Home` ribbon, select resize and set the image size to 128 x 128 pixels. Feel free to adjust the zoom level in the lower right part of the window. Select the `Brush` tool, chose the largest line size available and black color. Draw your favourite digit (0-9) and save it in the `GIF` format. Make sure that your hand-drawn digit fills the canvas and is nicely centred with roughly 20 pixels margin to the top, bottom, left and right.

b) Use `Pillow` to load your hand-drawn number into a Jupyter notebook via:

```
img = Image.open(filename).convert('L')
```

Call the rescaling method

```
imgRe = img.resize(size=(imgSize,imgSize))
```

to downscale your image to 28 x 28 pixels.

c) Use plt.subplots to generate a single plot of five MNIST images of your digit. Discuss differences and similarities between your handwriting and the MNIST sample.

Problem 3

In the lecture, we have established the foundation of a neural network class `neuralNetwork` which will be extended method by method throughout the next weeks before we switch to `Tensorflow`.

a) Copy all the code that is required to create an instance of the class `neuralNetwork`.

b) Write a function `softmax(arr)` that applies the softmax operation to a Numpy array:

$$\text{softmax}(x) = \frac{\exp(x_i)}{\sum \exp(x_i)}$$

The softmax operation is commonly used to normalize the output layer in multi-class models to get a probabilistic vector.

c) Use the network's `query` method to feed your handwritten digit into the network. Divide the image array by 255 before you pass it to the query method. Apply the `softmax` function to the output vector and verify that the sum of its entries equals 1.