Medical Data Science, WS 2023/2024

Prof. Dr. Nico Pfeifer

17.01.2024

Oğuz Ata Çal 6661014

Edward Beach 5451904

# Assignment 5

## Problem 1

**a)** One improvement is the increase in computational power (deep learning algorithms are usually computationally demanding and older hardware would have been too inefficient). Another factor is the rise of big data, we previously did not have many large (labeled) data sets that are useful for deep learning methods. Another factor is the improvement of Neural Network architecture, e.g. introducing CNNs for images, developing new algorithms for optimization, etc.. Some uses of deep learning are computer vision, object detection, classification, and medical image analysis.

**b)** In sparse encoding, data is represented as a linear combination of a set of basis functions. Sparse encoding optimizes by trying to find a sparse set of coefficients to represent the data where only a few elements are non-zero. This can be used for dimensionality reduction, compact feature representation, or signal denoising

**c)** The encoder transforms input data into a compressed representation with the most important features of the data (i.e. a type of dimensionality reduction). The decoder tries to reconstruct the original input data from the transformed representation that it received from the encoder. It is useful since we can extract only the most important features from our data ignoring unnecessary computation or distracting noise. We can avoid identity mapping by using regularization techniques or purposefully adding noise to the input data.

**d)** Activation functions are usually non-linear functions that calculate the output of a given neuron in a neural network. An example would be the Rectified Linear Unit (ReLu). The function is defined as $f(x) = \max(0,x)$. Thus, if the input into the neuron is negative the result will always be 0 and the value x otherwise.

**e)** A 2-D contact map in the context of amino acids is a type of matrix that represents the spatial proximity of given amino acids. The rows and columns correspond to the amino acids and the matrix values indicate if those amino acids are spatially in contact with each other, allowing us to gain insights into the 3-D structure of the protein.

**f)** Amino acids can be represented by 25-dimensional vectors. 20 values represent the evolutionary information for each amino acid type, 3 binary values represent the predicted secondary structure the amino acid is in and 2 binary values represent the predicted accessibility of the amino acid in the 3-D structure

**g)** L is the length of the protein sequence and L/5 is the top-scored pairs of amino acids we would analyze for performance measure. The L/5 contract pairs are compared to the real contacts of the amino acids and the rate of false positives/false negatives can be calculated, which in turn allows us to evaluate the performance.

**h)** One method to measure dependence between two variables is by calculating their correlation. If the correlation is 0 they are independent, if they are between [-1,1]\{0} they are dependent.

**i)** Max pooling is used to reduce the dimension of the spatial information in the input data. It selects the max value from neighboring values (e.g. pixels in images) in the input data. It is a procedure with information loss since we lose the non-max values and thus lose some dimensions of the input data.

## Problem 2

**a)** By reading the probabilities off the joint distribution table: P(c) = 0.04 + 0.23 + 0.04+ 0.27 = 0.58

**b)** P(d) = 0.06 + 0.17 + 0.04+ 0.27 = 0.54

**c)** P(d,c) = 0.04 + 0.27 = 0.31

**d)** P(s|d,c) = 0.27

**e)** Value for normalizing: 0.04+0.23+0.04+0.27=0.58

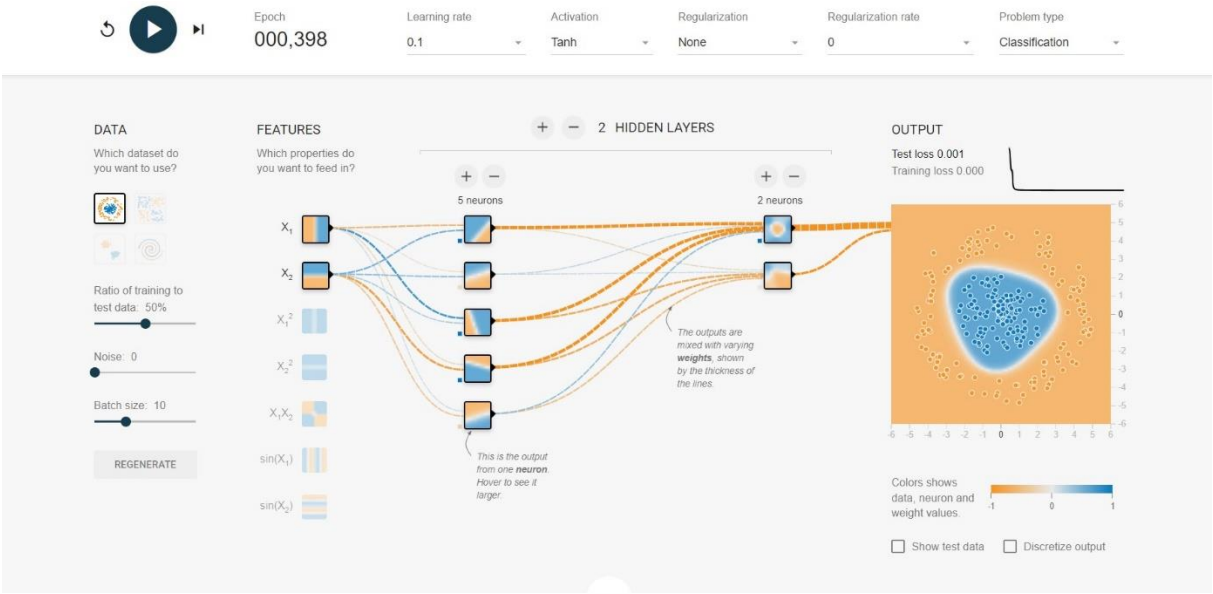
| d | **c** | **s** | P(d,c,s\|do (c)) |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 0.04/0.58 = 0.069 |
| 0 | 1 | 1 | 0.23/0.58= 0.4 |
| 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 0.04/0.58= 0.069 |
| 1 | 1 | 1 | 0.27/0.58 = 0.466 |

Value for normalizing: 0.06 + 0.13 + 0.06 + 0.17 = 0.42

| d | c | s | P(d,c,s\| not do(c) |
|---|---|---|---|
| 0 | 0 | 0 | 0.06/0.42 = 0.14 |
| 0 | 0 | 1 | 0.13 / 0.42 = 0.31 |
| 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0.06/0.42 = 0.14 |
| 1 | 0 | 1 | 0.17/0.42 = 0.4 |
| 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 0 |

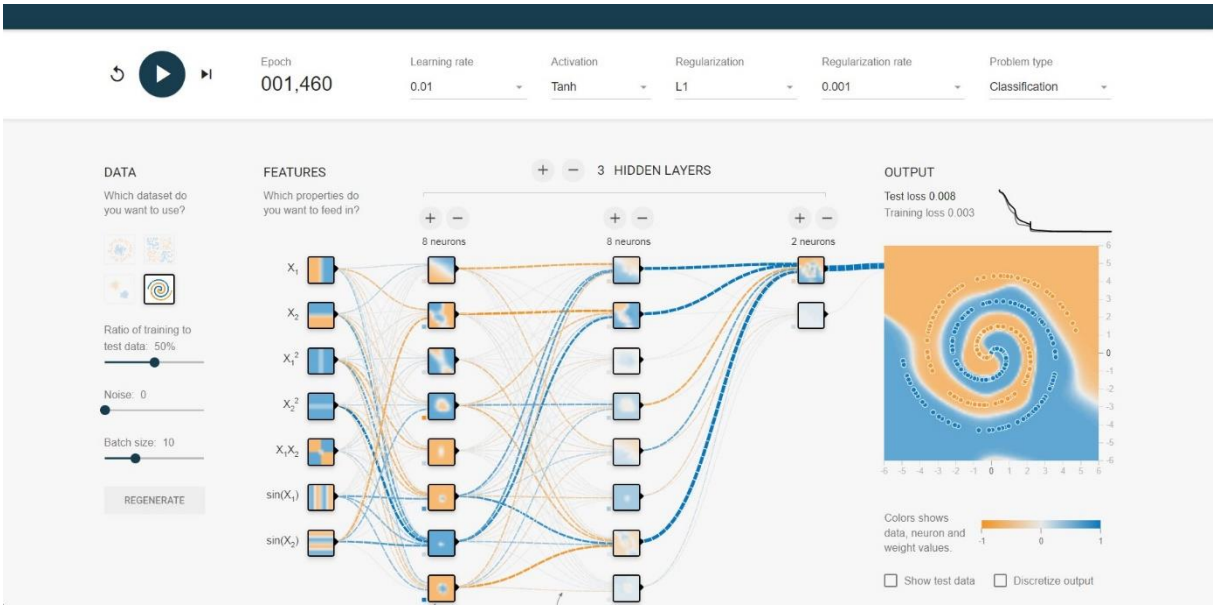
## Problem 3

**a)** The screenshot showing 0.001 Test loss:




**b)** There are 2 hidden layers, the first with 5 neurons and the other with 2 neurons and only the first two features are used. The decision surface is learned mostly by the first 5 neurons. As can be seen from the photos, each of them encapsulates a different line and all of them combined make the circular border.

**c)** The screenshot showing 0.008 Test loss:




d) There are 3 hidden layers with 8 neurons each in the first two layers and 2 neurons on the last layer. All features are used. My thought process was since this is a rather complex decision boundary, make the system as complex and see if it achieves less than 0.05 error. That is why all features are used and there are many neurons. L1 regularization was also needed to lower the test/training loss. Because the architecture is complex, we cannot semantically understand from the pictures which neuron learns which feature.