PROJECT REPORT

Artificial Neural Networks

Martin Trejtnar (er1583)

25th May 2020

Obsah

Seznam obrázků

1 Introduction

This is the report concerning the Artificial Neural Networks class project. Project is implemented in the Anaconda Python distribution, which is being used for scientific computing including machine learning application and large-scale data processing.

The task is to build and train a neural network from scratch using the knowledge and techniques acquired during the semester.

2 Data set description

Project is based on the `Glass Identification Data Set`. Data set is classifying the types of glass. The motivation comes from the criminal evidence recognition. The data set has 11 columns. First one is the attribute ID, followed by 9 chemical elements and the last one is the target.

- **1** Id number: 1 to 214
- **2** RI: refractive index
- **3** Na: Sodium (unit measurement: weight percent in corresponding oxide, as are attributes 4-10)
- **4** Mg: Magnesium
- **5** Al: Aluminum
- **6** Si: Silicon
- **7** K: Potassium
- **8** Ca: Calcium
- **9** Ba: Barium
- **10** Fe: Iron
- **10** Type of glass: (class attribute)

The data set contains from 214 instances of real numbers. There are no missing values.

3 List of used libraries

- **neurolab** - based neural networks, training algorithms
- **matplotlib.pyplot** - for plotting results
- **sklearn.preprocessing** - for changing, respectively scaling raw features
- **train_test_split** - for splitting data between training and testing part
- **pandas** - for loading data from file
- **numpy** - arrays, containers

Try	Epochs	Layers	Neurons	Test size	Epoch 100	Last epoch
1	3000	1	10	0.30	21.39417144038844	9.68317086097056
2	**5000**	**1**	**10**	**0.30**	**21.21510795970379**	**8.249118989413011**
3	10000	1	10	0.30	22.212276123119764	7.874884337082903
4	20000	1	10	0.30	22.090396177815457	8.36855479414207

4 Optimizing the neural network - unsupervised

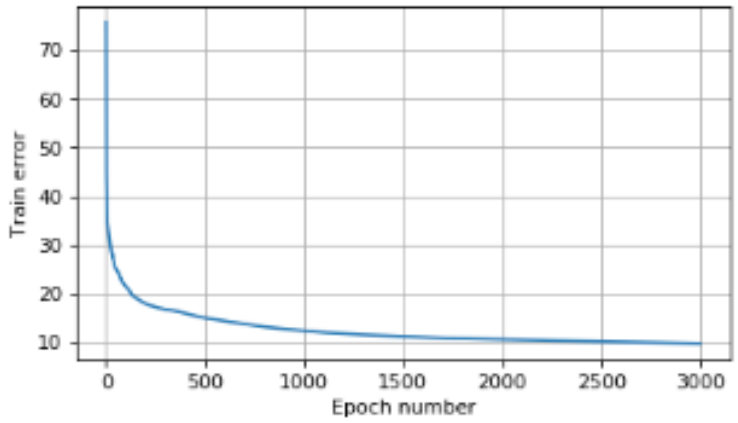
4.1 Changing the number of epochs

Settings:

- **Split rate** - 70% training, 30% testing
- **Hidden layers** - 1
- **Number of neurons** - 10

4.1.1 3000 epochs

[1] [2] [3]



Obrázek 1: Scatter 3000 epochs

4.1.2 Table of tests

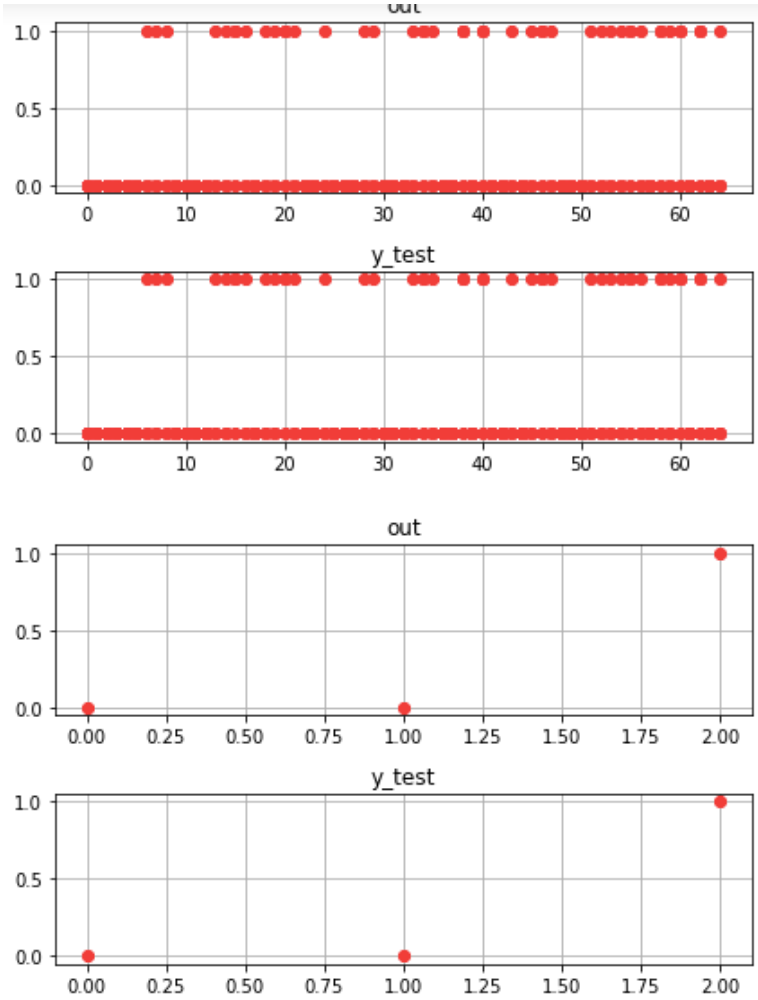
4.1.3 Chosen settings - 5000 epochs

The error difference between 5000 a 10000 epochs is not that significant, so I have decided to choose the amount of 5000. [4] [5]

4.2 Changing the test size

Settings:

- **Number of epochs** - 3000
- **Layers** - 1
- **Number of neurons** - 10



Obrázek 2: Error graph 3000 epochs

4.2.1 Tried settings

[6] [7] [8] [9] [10]

4.2.2 Chosen settings

Regarding the plots the best result seems to be given when the testing part equals **50 percent**.

4.3 Changing number of layers and neurons

5 Table of tests

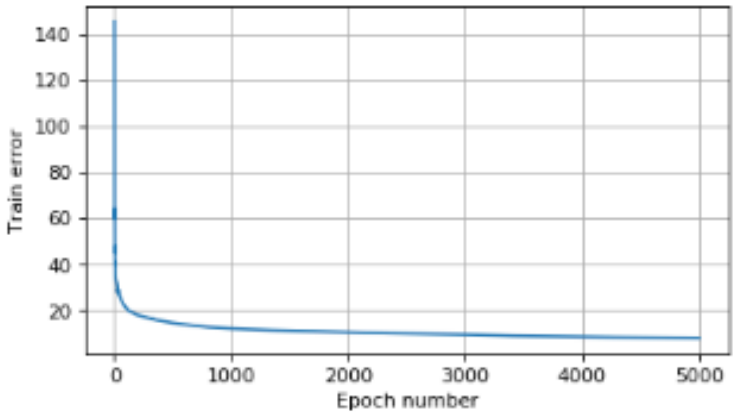
5.1 Chosen settings - 10 neurons in both 2 layer

6 Supervised learning

The results I have got are unfortunately disappointing. The train centers are for some reason way too far away from the train samples. It might be caused by the given data. I did not manage to achieve any satisfying plots.

```
Epoch: 100; Error: 21.39417144038844;
Epoch: 200; Error: 17.893373617336874;
Epoch: 300; Error: 16.681439086830625;
Epoch: 400; Error: 15.900120666403218;
Epoch: 500; Error: 14.918372701926646;
Epoch: 600; Error: 14.27484782466632;
Epoch: 700; Error: 13.687475764938064;
Epoch: 800; Error: 13.096734744080775;
Epoch: 900; Error: 12.672219574436955;
Epoch: 1000; Error: 12.27252204213126;
Epoch: 1100; Error: 11.935329228970794;
Epoch: 1200; Error: 11.67844271399852;
Epoch: 1300; Error: 11.47502375196699;
Epoch: 1400; Error: 11.277329561250596;
Epoch: 1500; Error: 11.091937705358099;
Epoch: 1600; Error: 10.95170026374591;
Epoch: 1700; Error: 10.79630980052773;
Epoch: 1800; Error: 10.674159044240747;
Epoch: 1900; Error: 10.565502742900234;
Epoch: 2000; Error: 10.470846506140244;
Epoch: 2100; Error: 10.382718483464448;
Epoch: 2200; Error: 10.299414877949493;
Epoch: 2300; Error: 10.224492377360004;
Epoch: 2400; Error: 10.146792322993393;
Epoch: 2500; Error: 10.0754707994948;
Epoch: 2600; Error: 10.007460900878044;
Epoch: 2700; Error: 9.920199330010595;
Epoch: 2800; Error: 9.847637324819086;
Epoch: 2900; Error: 9.767417127008915;
Epoch: 3000; Error: 9.68317086097056;
The maximum number of train epochs is reached
```

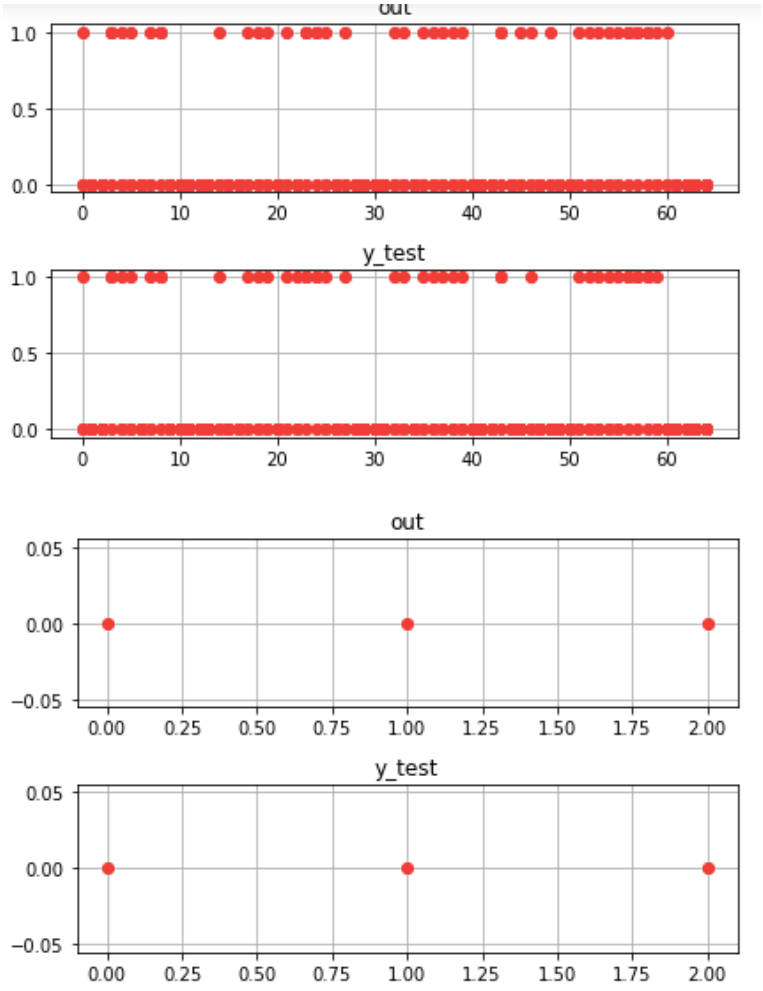
Obrázek 3: Table of error 3000 epochs



Obrázek 4: Error 5000 epochs

7 Essential part of constructing a neural network

The most important part in my opinion is the data given. Without proper data is impossible to build a well working neural network. Its necessary to handle missing values. If the data are fine a think that the thing



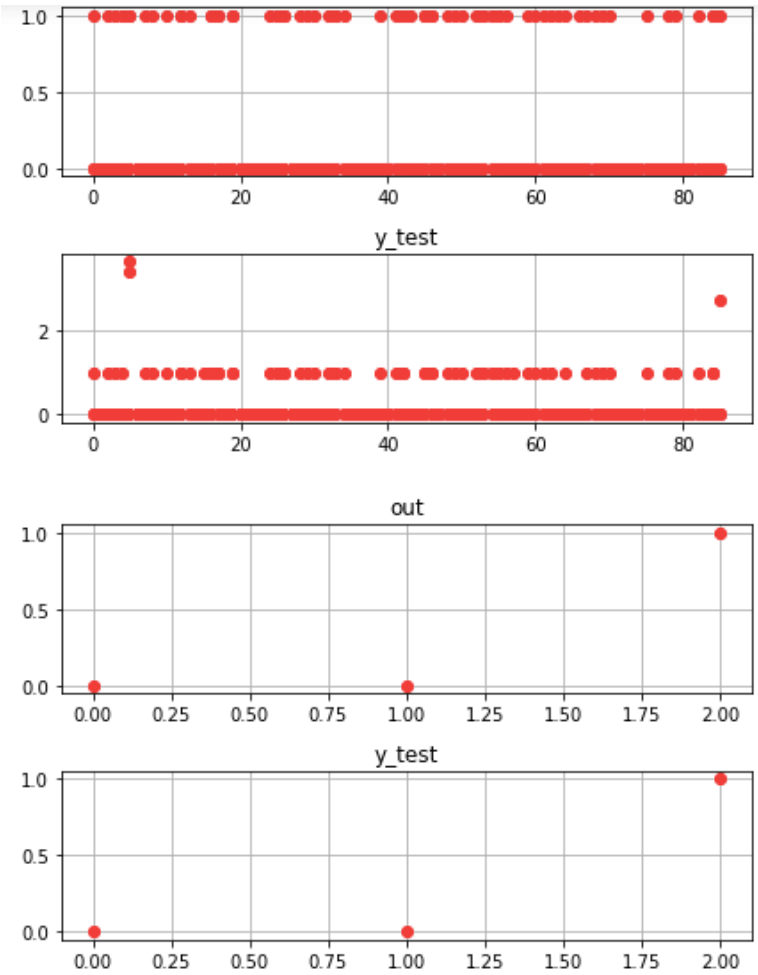
Obrázek 5: Scatter 5000 epochs

affecting the results the most is a choice of suitable number of layers and the amount of neurons in those.

8 Conclusion

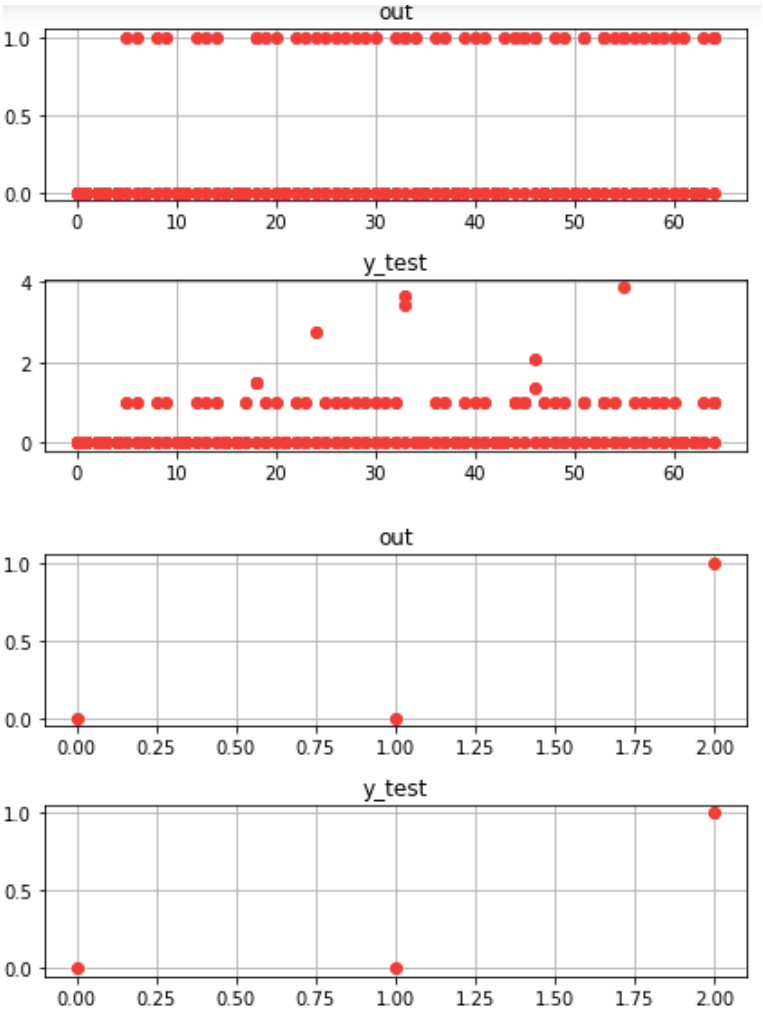
Regarding the tested settings the best results are given when I use 2 hidden layers with 10 neurons in both, with 5000 epochs and 50% testing part. With these settings i have achieved the error of **2.0001952145943953**. [12] [13] [14]

Also I have found out that on mine Ubuntu 18.04 there might some problem in the computation. Sometimes I have got very strange error values even though I was using the same settings. That might be caused by different architecture.

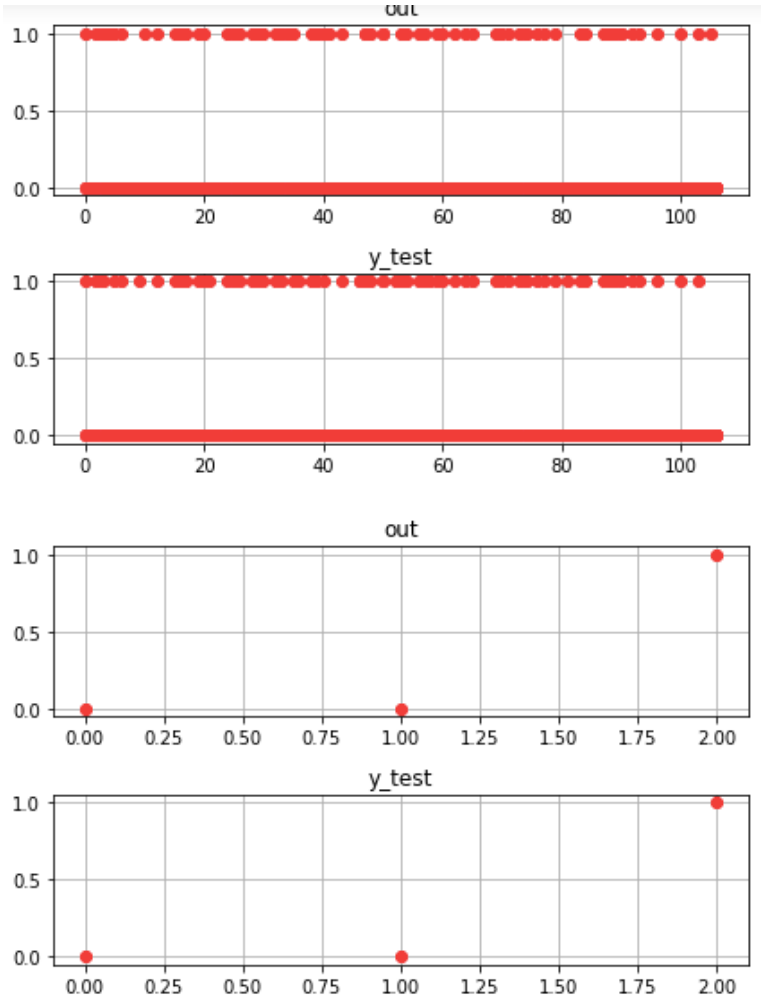


Obrázek 6: Scatter testing 40%

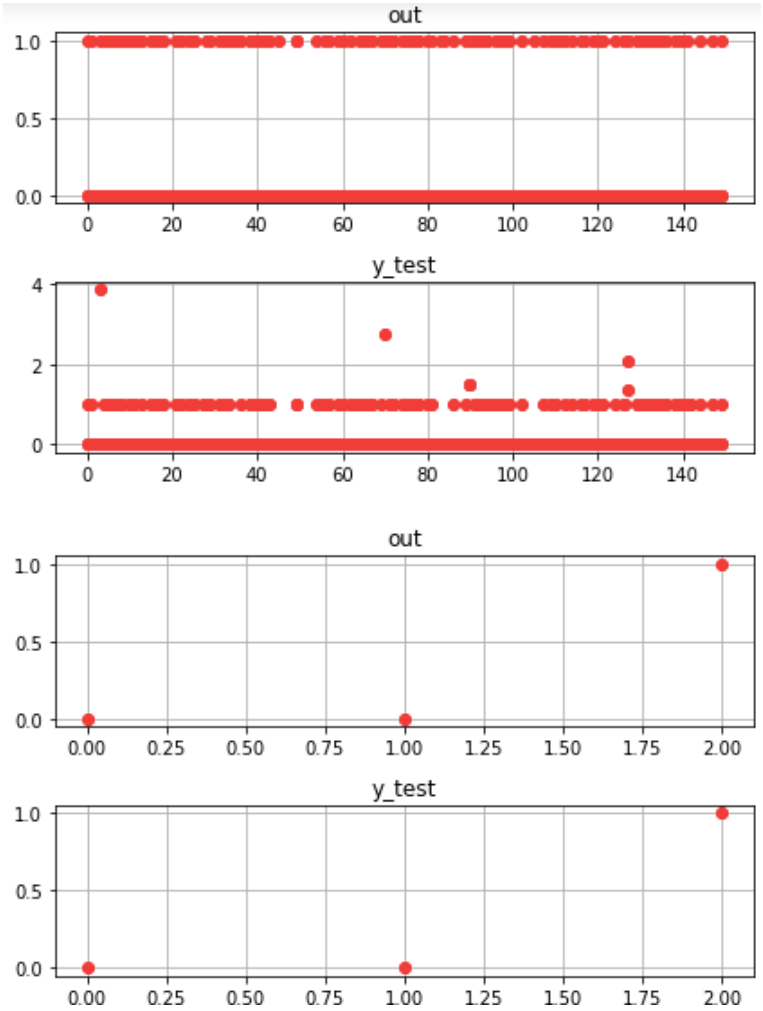
Try	Epochs	Layers	Neurons	Test size	Epoch 100	Last epoch
1	3000	1	10	0.30	21.39417144038844	9.68317086097056
2	3000	1	15	0.30	21.417002298166736	5.5816569423536215
3	3000	1	20	0.30	19.793526723631935	3.6331074596584085
4	**3000**	**2**	**10+10**	**0.30**	**21.69314972183241**	**2.296433373015625**
5	3000	3	15+10+10	0.30	23.52971282538101	8.5
6	3000	3	5+15+15	0.30	25.385522786093937	12.765774593631583



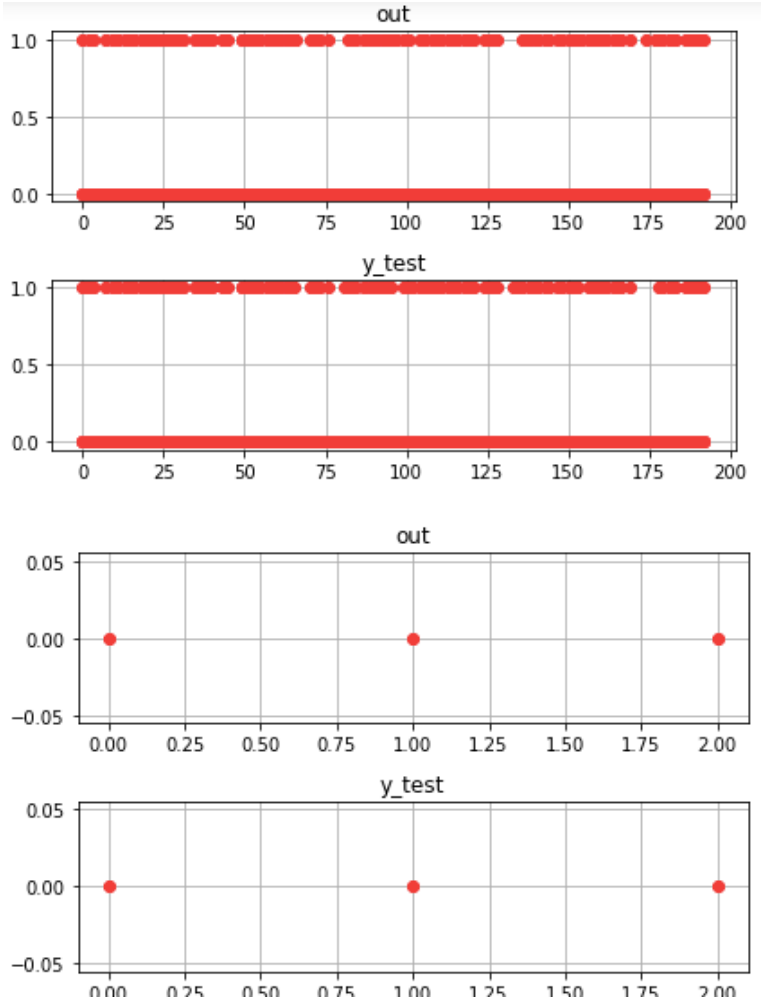
Obrázek 7: Scatter testing 30%



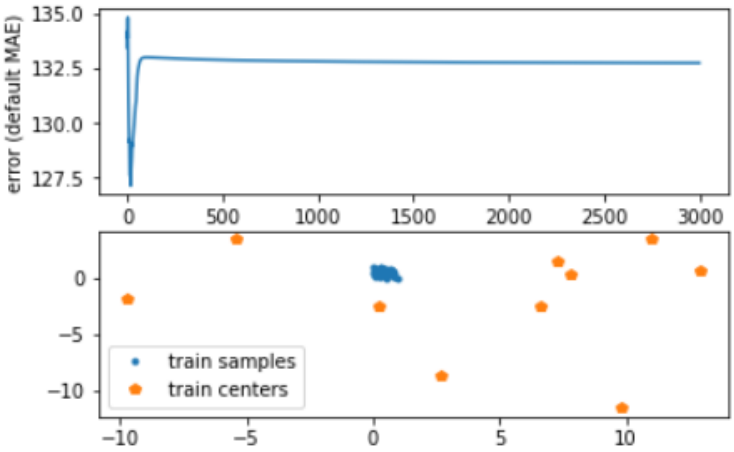
Obrázek 8: Scatter testing 50%



Obrázek 9: Scatter testing 70%



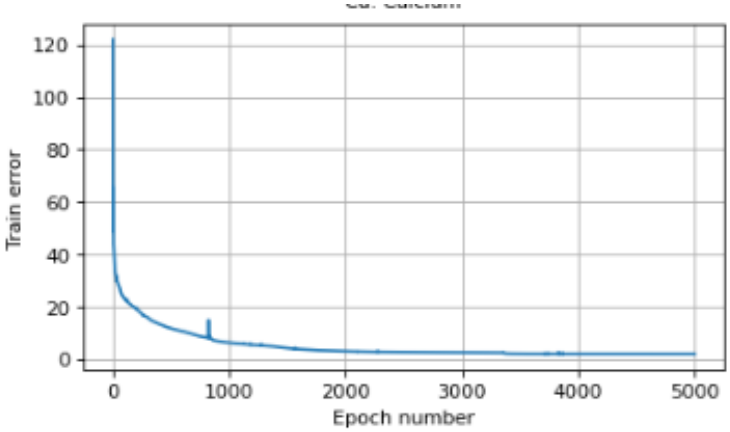
Obrázek 10: Scatter testing 90%



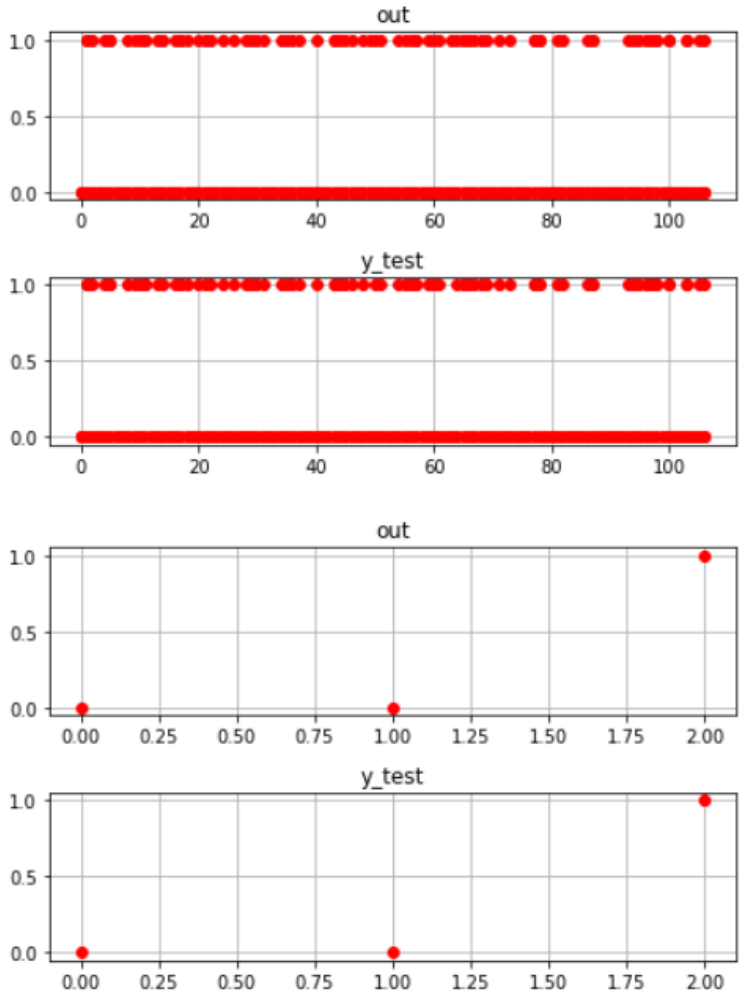
Obrázek 11: Kohonen Network

```
Epoch: 4600; Error: 2.0007510967310056;
Epoch: 4700; Error: 2.0005691476090446;
Epoch: 4800; Error: 2.0004175188009845;
Epoch: 4900; Error: 2.000255879615581;
Epoch: 5000; Error: 2.0001952145943953;
The maximum number of train epochs is reached
```

Obrázek 12: Minimal achieved error



Obrázek 13: Minimal achieved error - graph



Obrázek 14: Scatter of the best solution