Artificial Neural Networks

Final Project (Spring 2019)

Vojtěch Šikula – er1455

Training feed forward neural network to classified data from Haberman Survival dataset.

The Haberman Survival dataset contains 306 samples, and 4 atributes.

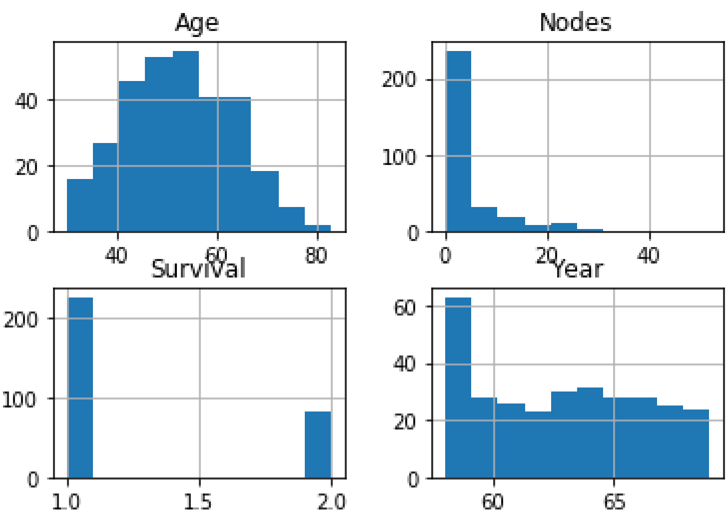
Attribute Information:

1. Age of patient at time of operation (numerical)
2. Patient's year of operation (year - 1900, numerical)
3. Number of positive axillary nodes detected (numerical)
4. Survival status (class attribute)

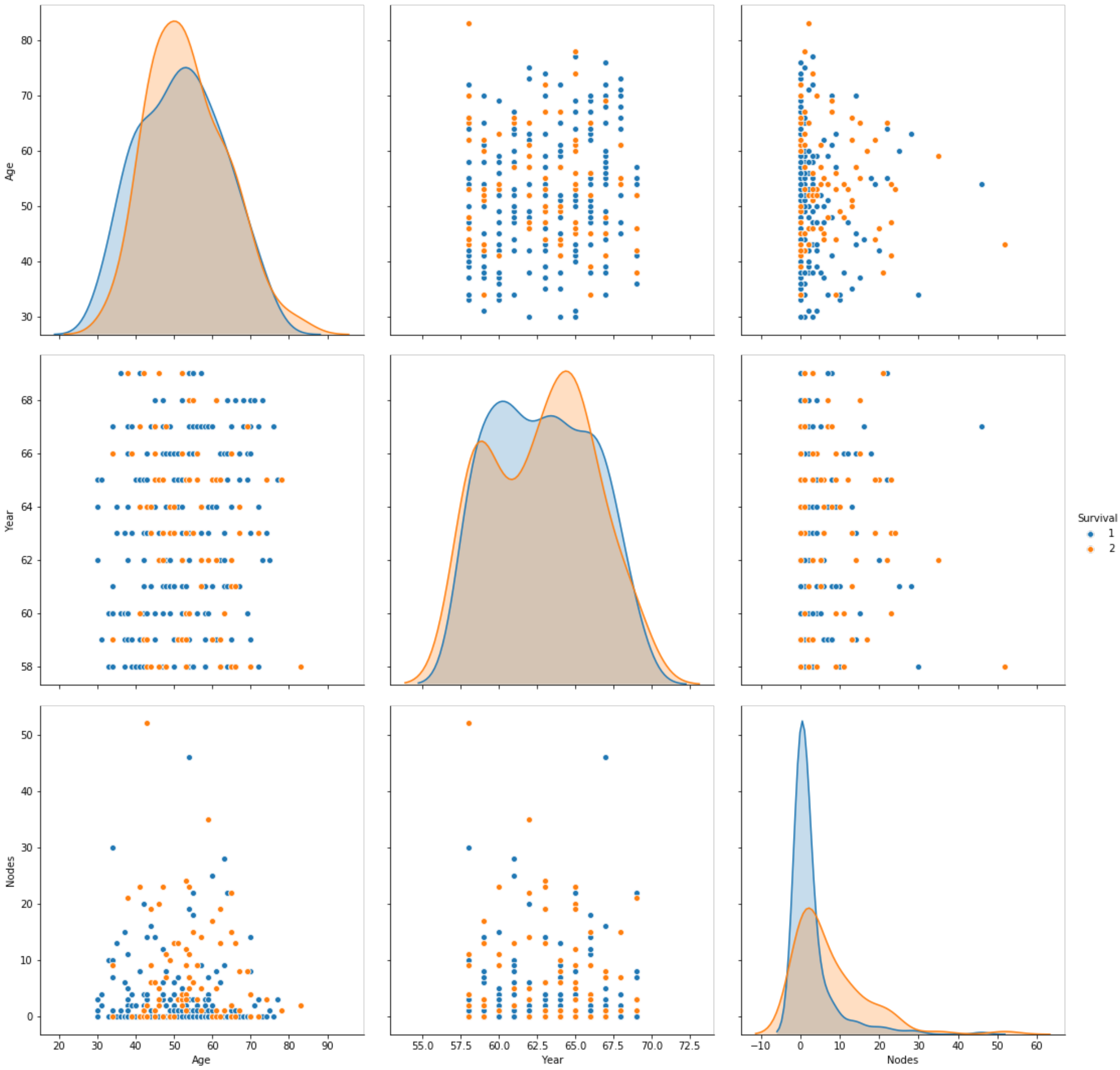
   1 = the patient survived 5 years or longer

   2 = the patient died within 5 year

Histogram of all attributes shows that attributes Age looks like normal distribution,majority of patients had small number of detected nodes, most patients survived more than 5 years and that similar amount of operation was done each year with exception of first year of survay.

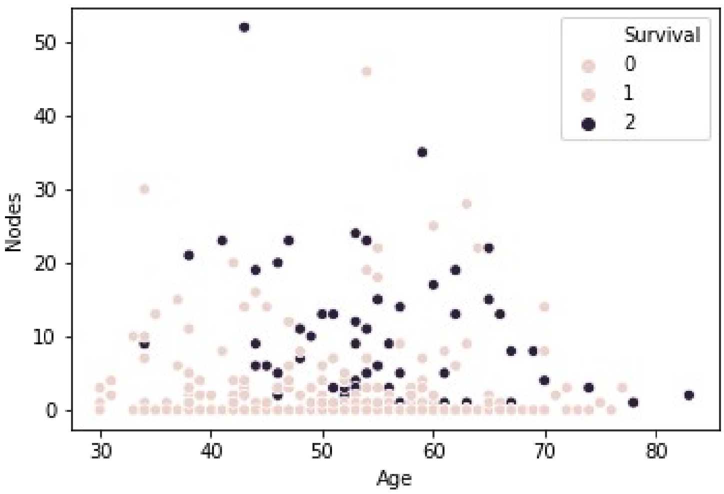


Next picture show pair plot of all attributes



To the next study was remove attribute “Year of opearation” by assumption that Survival is not dependent of year of operation.

Scatter plot of attributes Age and Nodes



The data was normalized, so that values of every attributes is between [0,1], by given formula:

$$X_n = \frac{(X - Xmin)}{(Xmax - Xmin)} * (-1) + 1$$

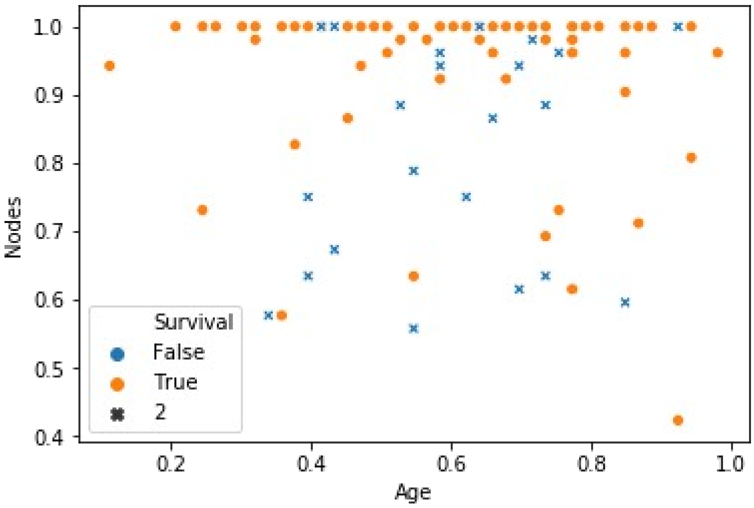
Where:

X – value that should be normalized
Xn – normalized value
Xmin – minimum value of X
Xmax – maximum value of X

First neural network

After spiting data into testing and training sets the first neural network was build with two hidden layers each with 10 neurons.

The NN has 0.74 accuracy on training set and 0.71 on testing sets. But this nn is strongly Bayes towards classifying into class 1 – that the given patients survived more than 5 years. This is probably result of uneven distribution of classes.



Results

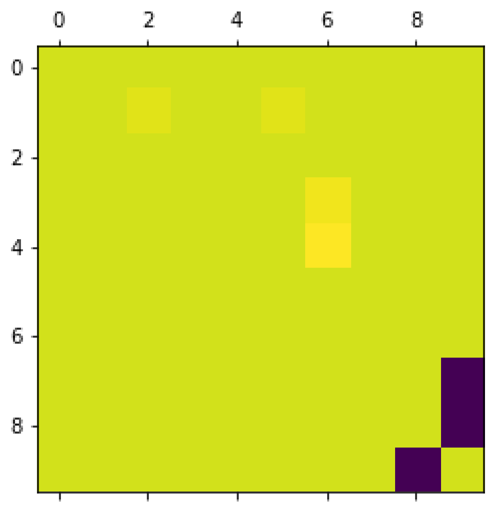
Different architectures

The neural network with one, two and three hidden layers was build and trained with every combination of 10,40,70,100,130,160,190,220,250,280 neurons in hidden layer. The learning rate was decrease to 0.05 because of smoother learning.

1 hiddne layer – usualy same result as above with small improvement with 220 or 250 neurons in hidden layer.

Neurons in hidden layer	accuracy
10	0.7174
40	0.7174
70	0.7174
100	0.7174
130	0.7174
160	0.7174
190	0.7174
220	0.7283
250	0.7283
290	0.7174

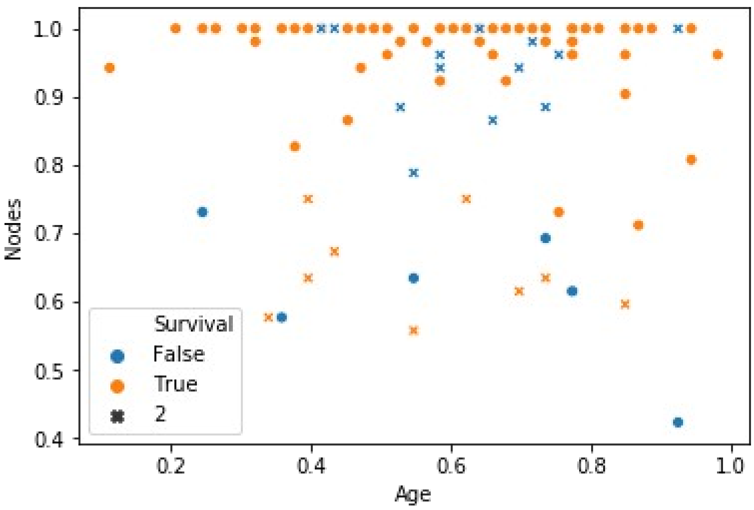
2 hidden layers – brighter color means better accuracy. Best accuracy at index (4,6) → (130,190) neurons in hidden layer, with accuracy 0.75. Dark collor is accuracy 0.2826, where the neural network classified all samples into class 2 → died within 5 years.



3 hidden layers – does not give better results.

```
best found accuracy:
('1 hidden layer', 0.7282608695652174)
('2 hidden layers', 0.75)
('3 hidden layers', 0.7282608695652174)
```

Scatter plot of best result – accuracy 0.75:



Different split rate of testing and training data:

Test-Train rate	Accuracy
10 – 90	0.64516
20 – 80	0.70967
30 – 70	0.75
40 – 60	0.26829
50 – 50	0.76470
60 – 40	0.7391
70 – 30	0.7441
80 – 50	0.2653
90 – 10	0.7282

In general we should see, that bigger training set should give better results – nn has more data to learn from. But this is hart to show on such a small dataset as this, where is only couple hundreds samples.

Conclusion

The most important thing to successfully train neural network are good data – sufficiently big data-set with minimum noisy samples. Training neural network is in most parts experimental job, where is almost impossible to ques how good can be without running the experiment.