University of Siena
MSc Artificial Intelligence

Machine Learning Assignment

Natasa Brisudova

2021/2022

Early initiation of targeted treatment can prevent possible irreversible neurological complications of spondylodiscitis (SD) and/or spinal metastases (MET), but differentiation may be a diagnostic problem, especially in the early stages. Aim of this project is to build a neural network to distinguish SD and MET given the 31 radiomic features from lessions and discuss the efficiency of the model.

1 Introduction

The whole data set used for this project is divided into two subsets:

- **Test and train set** - used to build the neural network and to optimize the hyperparameters to get the best possible model consisting of 80 lessions (40 of SD and 40 of MET)
- **Validation set** - used to test the neural network to avoid possible over-fitting consisting of 91 lessions (55 of SD and 36 of MET)

The feedforward neural network (Fig.1) is described as follow:

- 31 input units (for each feature)
- 1 output unit with the *sigmoid* activation function $S(z) = \frac{1}{1+e^{-z}}$ (Fig.2):

with

- **A)** 1 hidden layer with 6 units with the *tanh* activation function: $\tanh(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}}$ (Fig.3)
- **B)** 2 hidden layers with 7 and 3 units respectively with the *tanh* activation function: $\tanh(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}}$ (Fig.3)

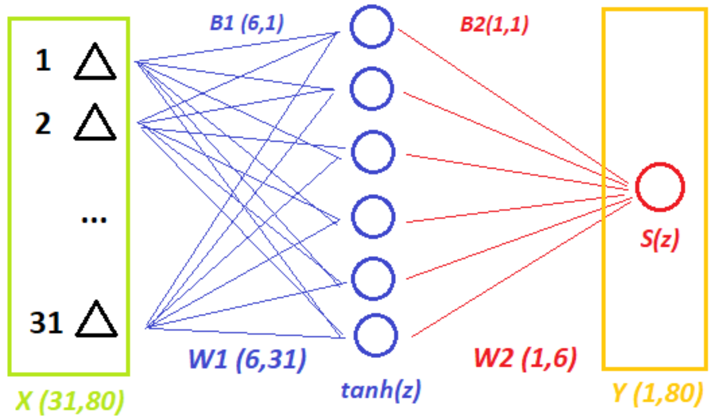

Figure 1: Scheme of the neural network A (with one hidden layer) with data sets and parameters with dimensions (n, m) where n are rows and m columns: X = inputs of train and test set; W = weights; B = bias; Y = outputs of train and test set.

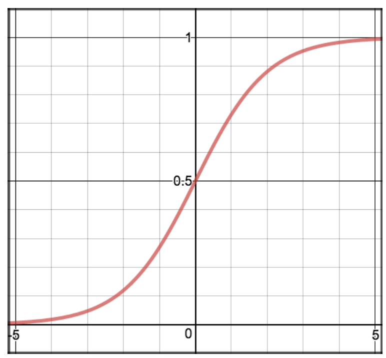

Figure 2: Sigmoid takes a real value as input and outputs another value between 0 and 1. It's easy to work with and has all the nice properties of activation functions: it's non-linear, continuously differentiable, monotonic, and has a fixed output range.[1]

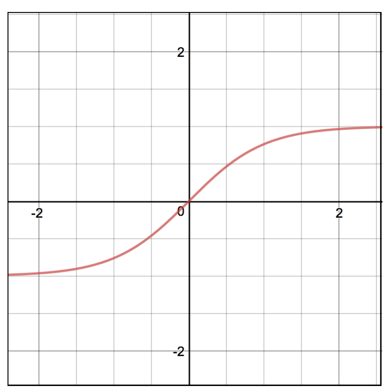

Figure 3: Tanh squashes a real-valued number to the range [-1, 1]. It's non-linear and its output is zero-centered. Therefore, in practice the tanh non-linearity is always preferred to the sigmoid nonlinearity.[1]

2 Methods

The performance of the neural network was tested changing the following:

- learning mode protocols
  - **batch mode:** for each epoch the parameters are updated just once
  - **stochastic (on-line) mode:** the parameters are updated after every training example, i.e n times for each epoch
  - **mini-batch mode:** every epoch the parameters are updated n/b times where n is the number of training examples and b the size of the mini-batch.
- number of iterations (epochs)
  - 5000
  - 10000
  - 20000
  - 50000
- learning rate: from 0.001 to 0.01

3 Code explanation

All the references are shown in the script.

The code is divided into 3 parts:

1. **Loading and normalizing data sets** (so that all the values range from 0 to 1)
2. **Defining functions for:**
   - initializing the parameters W and b randomly
   - activation functions: *tanh* (Fig. 3), *sigmoid* (Fig. 2)
   - forward propagation
   - cross-entropy cost function
   - backward propagation
   - updating parameters:
     - batch gradient learning mode (mini-batch size = 80)
     - stochastic (on-line) learning mode (mini-batch size =1)
     - mini-batch gradient learning mode (mini-batch size = 16)
   - prediction
3. **Final model and printing the results**

4 Results

All the references including the graphs and accuracy tables can be seen in the excel file.

4.1 Results of NN with 1 hidden layer (A)

In the neural network with the **batch gradient** learning mode (Fig.4) the learning rate 0.0075 seems to be the most suitable. In this case, the more iterations the better performance, therefore the run with 50000 iterations had the highest accuracy (Fig.10).

The neural network with the **stochastic (on-line)** learning mode (Fig.5), as expected, did not perform so well, caused by the smaller data set (80 lessions). The learning rate of 0.005 seems to be slightly better than 0.0075, however in this case the highest number of iterations equal to 50000 started to cause over-fitting (the accuracy of the model increased while the accuracy of validation set decreased a lot). Because of that, the neural network with learning rate 0.005 and 20000 iterations fits the best (Fig.11).

In the neural network with the **mini-batch** learning mode (Fig.6) the mini-batch size of b=16 was used considering the data set size. The model performed better with higher learning rates as 0.0075 and 0.01, and similarly, with the iterations equal to 50000 the accuracy of validation set significantly decreased caused by the over-fitting of the model. Therefore, I would choose the model with the learning rate 0.0075 and iterations equal to 20000 (Fig.12).

BATCH GRADIENT		*iterations*							
accuracy - validation & model		5000		10000		20000		50000	
learning rate	0,005	88,17	78,75	90,32	83,75	92,47	86,25	95,69	90
	0,0075	97,84	91,25	97,84	91,25	94,62	93,75	94,62	96,25
	0,01	90,32	83,75	92,47	86,25	95,69	88,75	97,84	90

Figure 4: Accuracy comparison of neural network with the batch gradient learning mode

STOCHASTIC GRADIENT		*iterations*							
accuracy - validation & model		5000		10000		20000		50000	
learning rate	0,001	70,96	76,25	68,81	76,25	70,96	76,25	90,32	90
	0,005	72,04	76,25	90,32	90	93,54	95	91,39	98,75
	0,0075	72,04	76,25	93,54	92,5	91,39	93,75	91,39	98,75

Figure 5: Accuracy comparison of neural network with the stochastic (on-line) gradient learning mode

MINIBATCH GRADIENT		*iterations*							
accuracy - validation & model		5000		10000		20000		50000	
learning rate	0,001	88,17	78,75	90,32	83,75	92,47	86,25	95,69	90
	0,005	94,62	86,25	95,69	90	97,84	90	94,62	91,25
	0,0075	95,69	87,5	97,84	91,25	97,84	91,25	94,62	93,75
	0,01	95,69	90	97,84	90	96,77	91,25	95,69	96,25

Figure 6: Accuracy comparison of neural network with the mini-batch (b=16) gradient learning mode

4.2 Results of NN with 2 hidden layers (B)

In the neural network with the **batch gradient** learning mode (Fig.7) the learning rate 0.01 seems to work the best. With iterations equal to 50000 a slightly over-fitting can be spotted, however the decrease in the validation set accuracy is less significant, therefore we can assume it to be the best model (Fig.10).

The neural network with the **stochastic (on-line)** learning mode (Fig.8) was performing well only with the learning rate 0.001 and up to the iterations equal to 20000. However, the model with 5000 iterations fits the best. The runs with higher learning rates showed significant over-fitting of the data as the model accuracy reaching the highest possible accuracy meanwhile the accuracy of validation set keeps decreasing (Fig.11).

In the neural network with the **mini-batch** learning mode (Fig.9) the mini-batch size of b=16 was used considering the data set size. The performance did not change a lot with the learning rates higher than 0.005. The model with the best accuracy seems to be with iterations equal to 20000 (Fig.12).

BATCH GRADIENT		*iterations*							
accuracy - validation & model		5000		10000		20000		50000	
learning rate	0,005	81,72	75	91,39	87,5	96,77	92,5	98,92	93,75
	0,0075	89,24	85	95,69	88,75	97,84	91,25	97,84	93,75
	0,01	91,39	87,5	96,77	92,5	98,92	91,25	97,84	96,25

Figure 7: Accuracy comparison of neural network with the batch gradient learning mode

STOCHASTIC GRADIENT		*iterations*							
accuracy - validation & model		5000		10000		20000		50000	
learning rate	0,001	97,84	97,5	96,77	98,75	95,69	98,75	86,02	100
	0,005	96,77	98,75	91,39	100	90,32	100	90,32	100
	0,01	92,47	100	88,17	100	89,24	100	89,24	100

Figure 8: Accuracy comparison of neural network with the stochastic (on-line) gradient learning mode

MINIBATCH GRADIENT		*iterations*							
accuracy - validation & model		5000		10000		20000		50000	
learning rate	0,001	81,72	75	91,39	86,25	96,77	92,5	98,24	93,75
	0,005	97,84	92,5	98,24	93,75	97,84	96,25	96,77	98,75
	0,0075	97,84	92,5	97,84	93,75	97,84	98,75	95,69	98,75
	0,01	98,92	93,75	97,84	96,25	96,77	98,75	93,54	98,75

Figure 9: Accuracy comparison of neural network with the mini-batch (b=16) gradient learning mode

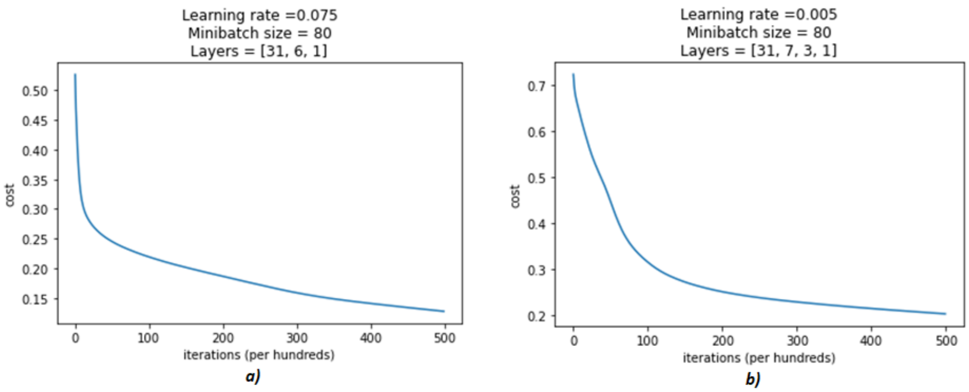

Figure 10: Number of iteration vs. cost function graph for NN with batch gradient learning mode with: a) 1 hidden layer (A); b) 2 hidden layers (B)

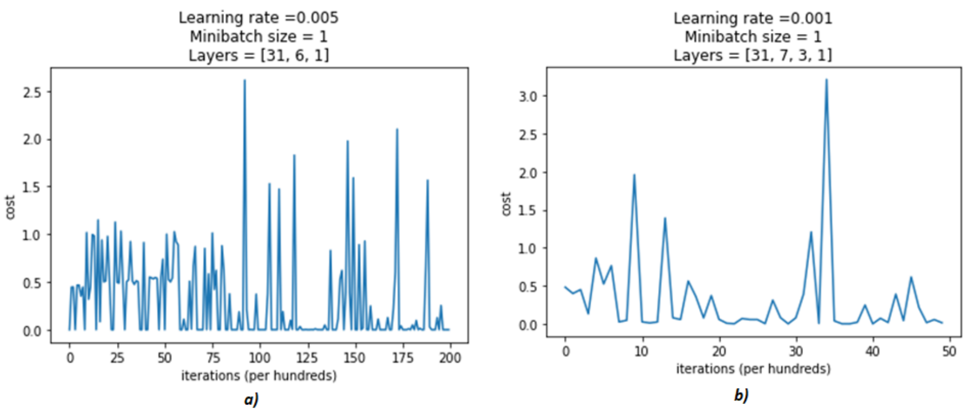

Figure 11: Number of iteration vs. cost function graph for NN with stochastic (on-line) gradient learning mode with: a) 1 hidden layer (A); b) 2 hidden layers (B)

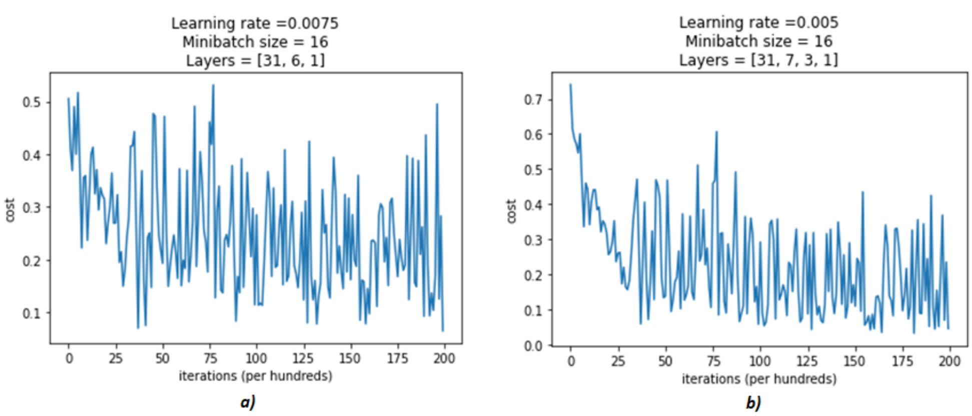

Figure 12: Number of iteration vs. cost function graph for NN with mini-batch gradient ($b = 16$) learning mode with: a)1 hidden layer (A); b) 2 hidden layers (B)

5 Conclusion

The neural network with stochastic (on-line) learning mode seemed to perform the worst, on the other hand the mini-batch gradient learning mode (with mini-batch size = 16) the best. The cost function of the neural network with the stochastic (on-line) learning mode (Fig. 11) does not follow the trend of decreasing, possibly because of the wrong chosen interval on the x-axis.

Overall, looking at the accuracy results of both of the neural networks (A and B) it can be confirmed that neural network can be used to distinguish spondylodiscitis and metastases, while the performance of the model with 2 hidden layers (B) is better than the one with 1 hidden layer (A).

References

[1] `https://github.com/bfortuner/ml-glossary.git`

[2] `NN_tanh_sigmoid.py`

[3] `results.xlsx`