Machine Learning for Bioinformatics

Exercise Sheet

Freie Universität Berlin, SoS 2024

Week 8 · Assignment on 05.06.2024. Submit until 14.06.2024 11pm.

Please note that the jupyter notebook must be submitted
along with the exercise sheet![1]

Name: Matriculation no.:

Name: Matriculation no.:

Artificial Neural Networks

Neural networks are complex models that are trained with gradient descent. PyTorch allows to easily combine individual layers to construct custom neural network models. A key feature of pytorch is that it allows you to compute derivatives (or gradients) for arbitrary functions, including neural networks. This feature is essential to train custom neural networks without having to manually define the gradients. In this exercise, we will dive into the automatic differentiation of pytorch and define a few simple models.

Assignment 1. *Introduction to PyTorch (optional)*

1. If you have never worked with PyTorch before you might want to go through the first assignment. It explains the most important concepts of the PyTorch library, such as tensors and automatic differentiation (autodiff). Points: 0

Assignment 2. *Numeric polynomial regression* Although ordinary least squares has a simple analytical solution, we use PyTorch to numerically compute the parameters of a polynomial regression model.

1. Complete the implementation of the *PolynomialNetwork* model. Points: 0

2. Complete the implementation of the *train* function. Points: 0

3. Run the optimization and report the resulting parameter vector. Points: 1

Model weights:

Assignment 3. *Spam classification* Similarly to the previous assigment, we implement here a simple classifier for predicting whether a text message is ham or spam. The model that we implement is a neural network with one layer and a sigmoid activation function, equivalent to a logistic regression model.

1. What is the number of ham and spam messages in the corpus? Points: 1

Ham: Spam:

[1] No points are awarded if an answer is only partially correct. Answers must be supported by results from the jupyter notebook.

2. Complete the implementation of the *DetectSpamNetwork* model. Points: 0

3. Complete the implementation of the *Trainer* class. Points: 0

4. After you have extracted the weights of your hidden layer, check what word is most indicative for ham and spam. Points: 1

   Ham Spam

5. Specify the values of the confusion matrix. Points: 1

Maximum number of points: 4