

# Semantic Computing

## Tutorial 8

Summer Semester 2018

Today the tutorial is a basic introduction to TensorFlow and RNNs. It is more about understanding code than programming. Please download the files in “tutorial8” from the github <http://github.com/dgromann/SemanticComputing/>.

### Exercise 1

*Basic operations in TensorFlow*

Please first read the exercise and look at the code before you run it.

- a) Take a look at “tutorial8\_1.py” and Exercise 1a. What do you think is the output of the three print statements?
- b) What will be the output of Exercise 1b? Uncomment and run it to see.
- c) What about Exercise 1c? What will the output be here?
- d) What about Exercise 1d? What will be the output here?
- e) Calculate the mean and the max value of a randomly generated array of numbers (variable: data) and output the result.
- f) Linear regression in TensorFlow:
  - Based on the input data, what is the expected output of this linear regression algorithm?
  - Create two float placeholders  $X$  and  $Y$ .
  - Create a variable  $w$  as a float with the name "weights".
  - Create a variable  $y\_model$  that multiplies  $X$  with  $w$ .
  - What kind of cost function is the provided cost function? What does the number in the brackets of the *GradientDescentOptimizer* mean?

### Exercise 2

*Understanding RNN*

Please open “tutorial8\_2.py”. This is a “walk-through” discussion of an implementation of RNNs from scratch without using any prepackaged libraries. We will discuss the code together in class to really understand what each element does. The objective of the code is to learn a character sequence that is able to predict meaningful text after being supplied with a character. In this case we will use tweets of Donald Trump.

- a) What is the activation function of the hidden layer used here?
- b) How many loops does the RNN take for each input sequence?
- c) How many updates does it have to go through in backpropagation with the unrolled RNN?
- d) Which loss function is used?

- e) Which optimizer is being used?
- f) Currently the loss is reduced very slowly, meaning the network does not learn really fast. Which hyperparameter do we have to change to improve the situation?
- g) Now that we have solved the slow improvement on the loss, how can we help the network to learn more refined representations of the tweets?