DL4NLP 2022 — Exercise 2

Steffen Eger,
Natural Language Learning Group (NLLG),
University of Bielefeld,
Summer Semester 2022

April 25, 2022

Task 1 (20min), Numeric weight optimization

Complement the code for numeric gradient adjustment given in *MLP_gradient_exercise.py*. Note: This is the same toy problem we have considered in Tutorial 1.

(i) What does the code in the functions 'gradient' and 'update' do?

(ii) Fix the code in 'YOUR CODE HERE'. Note that **three** pieces of code are missing.

(iii) Run the code for 10 random initializations of the weight matrices.

(iv) Which average, minimum and maximum accuracy do you get?

(v) What do you conclude from (iv)?

Task 2 (15min), n-gram language models

Implement a unigram, bigram and trigram language model completing the accompanying code *ngram.py*. 'Train' it on the accompanying texts 'Moby Dick' and 'William Shakespeare' (complete work).

(0) Fix the missing code in 'YOUR CODE HERE'.

(i) Generate language from both texts individually and across the n-grams.

(ii) What differences do you observe, between the texts and across the n-grams?

(iii) Are your generated sentences always full sentences?

Task 3 (10min), Symmetry of Joint Probability

For joint probability distributions, we have the property of symmetry:

$$P(A, B) = P(B, A)$$

Does this also hold for language models, which evaluate $P(w_1, w_2, \ldots, w_n)$?