Assignment 5: Classify Math Publications

AI-2 Systems Project (Summer Semester 2023)

Jan Frederik Schaefer

Friedrich-Alexander-Universität Erlangen-Nürnberg, Department Informatik

Topic:	Natural language processing, artificial neural networks
Due on:	October 20, 2023
Version from:	August 22, 2023
Author:	Jan Frederik Schaefer

1 Task summary

Train a neural network to suggest classifications for mathematical publications based on their titles. This could, for example, be used in a web interface where authors submit publications.

Didactic objectives

1. Learn the basics of a modern machine learning framework (pytorch),
2. get some hands-on experience with training a simple neural network for a natural language processing task.

Prerequisites and useful methods

1. Basics of neural networks,
2. basics of natural language processing.

2 Detailed task description

In recent years, natural language processing has advanced a lot through the use of sophisticated artificial neural networks (e.g. transformer models). These networks require a lot of computational power and are out of the scope of the AI lecture and the systems project. But simpler models can yield very good results as well! The goal of this assignment is to learn the basics of pytorch, a modern machine learning framework, by solving a text classification problem.

You can base your implementation on a pytorch tutorial for text classification [TC] and use the matrix rooms if you have any questions about the tutorial or pytorch in general. It

might also make sense to skim through the basic introduction tutorial for pytorch [PT]. It is possible to train a sufficiently good model within a few minutes (and without using a GPU).

2.1 The dataset

The dataset for this assignment is based on a publicly accessible dataset [ZBMd] from zbMATH [ZBM]. zbMATH collects abstracts and reviews of publications in the area of mathematics and its applications. For this assignment, we use the title of publications and their classifications, which are based on the Mathematics Subject Classification [MSC]. For simplicity, we only use the top-level classifications.

The training data consists of 1 000 000 titles, each of which has up to 5 classifications. They are stored in one file, where each line is a JSON object of the form

```
{"title": <TITLE>, "classifications": [<CLASSIFICATION 1>, ...]}
```

Here is an example entry:

```
{"title": "An extension of Chevalley's theorem to congruences modulo prime powers",
    "classifications": ["Number theory", "Group theory and generalizations"]}
```

There are also a test and a validation dataset that each list 10 000 publication titles (without classifications). The validation dataset is also available in the format of the training dataset, so you can use it for checking your solution.

The datasets are in the assignment repository [AR].

2.2 Evaluation

Your model should propose a set P_t of 5 classifications for every title t in the test dataset D. Your agent then gets the evaluation score

$$\frac{\sum_{t \in D} |P_t \cap C_t|}{\sum_{t \in D} |C_t|}$$

where C_t are the real classifications of t.

The proposed classifications should be stored in the same format as the training dataset:

```
{"title": <TITLE>, "classifications": [<CLASSIFICATION 1>, ...]}
```

You can use the validation dataset to evaluate your model (the assignment repository [**assignment**] also has a script for that). Unless you use the validation dataset for training, the evaluation score on the validation dataset and the test dataset should be very similar.

3 What to submit

Your solution should be submitted to your repository for this assignment. It should contain:

1. all your code,
2. a README.md file explaining how to run your code,
3. the proposed classifications for the test dataset,
4. a brief evaluation (what worked well, what was problematic?) either as a PDF file ($\approx$ 1 page) or as part of your README.md.

4 A few tips

1. The pytorch tutorial for text classification is a good starting point [TC], but it might also be worth looking at the general introduction to pytorch [PT].
2. There are many parameters you can change and (especially without much experience) it is hard to tell, what effect an adjustment has. It might be a good idea to take notes on the performance of different configurations to inform future experiments.
3. Here are a few ideas how you can change the network structure:
   (a) Add one (or more) layers on top and try out different activation functions (can make a big difference).
   (b) Increase/decrease the size of the embedding layer.
   (c) Try out different cost functions.
   (d) Try out different batch sizes and learning rates.
4. Another important aspect is the text pre-processing. Here are a few things you can experiment with:
   (a) Replace rare words with a token `<UNK>`. Rare words could be words that occur less than n times in the training data (for some value of n).
   (b) Ignore frequent words that are very unspecific (e.g. `"the"`). Such words are called "stop words".
   (c) Do not distinguish different forms of a word (e.g. `"groups"` and `"group"` should be treated as the same word). A tool for reducing a word to its base form ("stem") is called a stemmer. Many libraries for stemming exist.

5 Points

You can get up to 80 points for the evaluation score (see Section 2.2) of your model. Concretely, you will get the following points:

- 30 points if the evaluation score is ≥ 0.3
- 50 points if the evaluation score is ≥ 0.5
- 70 points if the evaluation score is ≥ 0.6
- 80 points if the evaluation score is ≥ 0.7

Note that it is possible to get a significantly better evaluation score than 0.7. Assuming you have at least a partial solution, you can additionally get up to 20 points for the quality of the submission (README, evaluation, ...). The maximum number of points is therefore 100. If the grading scheme does not seem to work well, we might adjust it later on (likely in your favor).

References

[AR] *Repository for Assignment 5: Classify Math Publications*. URL: `https://gitlab.rrze.fau.de/wrv/AISysProj/ss23/a2.5-classify-math-publications/assignment`.

[MSC] *Mathematics Subject Classification – MSC2020*. URL: `https://zbmath.org/classification/` (visited on 02/04/2022).

[PT] *PyTorch – Learn the Basics*. URL: `https://pytorch.org/tutorials/beginner/basics/intro.html` (visited on 08/01/2022).

[TC] *Text classification with the torchtext library*. URL: `https://pytorch.org/tutorials/beginner/text_sentiment_ngrams_tutorial.html` (visited on 08/01/2022).

[ZBM] *zbMATH Open*. URL: `https://zbmath.org` (visited on 02/04/2022).

[ZBMd] *zbMATH Open Dataset*. URL: `https://gl.kwarc.info/SIGMathLing/dataset-zbmath-open-2021` (visited on 02/04/2022).