Deep Learning in NLP: Semester outline

Christian Wurm & Tatiana Bladier

Heinrich-Heine-Universität Düsseldorf
Wintersemenster 2018/2019

Expectations from the course

You will be able to:

- ⋆ understand neural network models,
- ⋆ build neural network models using Python,
- ⋆ solve some NLP problems using deep learning models.

Motivation

- The knowledge learnt from Deep Neural Network (DNN) models is still largely untapped in the context of NLP.
- DNNs generalize well on unseen data.
- They are suitable to deal with outlying, missing, unstructured, and noisy data.
- Very flexible and can be paired with other techniques.
- Outperform linear models in classification and data analysis tasks.

Source: Younes Samih (2017)

What are we going to learn?

- Deep Learning needs five basic things (Chollet, 2017):

  → **Problem or task**: machine translation, sentiment analysis, text classification, part-of-speech tagging, text generation etc.

  → Input data points (**features**): files of people speaking, text files, images etc.

  → Examples of the expected outputs (**tags or labels**): transcripts of sound files, positive/negative, cat/dog/fish etc.

  → Suitable **algorithm to train** on those data.

  → **Way to evaluate** how good our algorithm is (distance between the algorithm's output and expected output).

Representation of the data → coming next week

- ⋆ Deep Learning is all about the **data**
- ⋆ Find suitable corpora online:
  - → different data formats (e.g. XML, raw text, HTML-documents, treebank formats etc.)
  - → `https://toolbox.google.com/datasetsearch`
- ⋆ Build own text corpora.
- ⋆ Prepare the data (preprocessing).
- ⋆ Provide appropriate representations for the input data.
  → data stored in Numpy arrays = *tensors*

Deep Learning Frameworks: Keras

- ⋆ Background information on different DL frameworks (e.g. Caffe, Torch, Pytorch, Keras).
- ⋆ Background information on Tensorflow and Theano (backend engines of Keras).
- ⋆ Introduction to Keras.
- ⋆ First deep learning project (predicting Boston housing prices).

Deep Learning tasks for NLP: Neural Architectures

Basic architecture types:

- ⋆ Recurrent neural networks.
- ⋆ Convolutional and pooling.
- ⋆ Recursive neural networks.

Deep Learning tasks for NLP: Neural Architectures

- ⋆ Document classification

  → identify the topic of an article or the author of a book.
- ⋆ Similarity comparisons

  → how closely related are two documents?
- ⋆ Sequence-to-sequence learning

  → decoding an English sentence into French.
- ⋆ Sentiment analysis

  → classify the sentiment of tweets or movie reviews as positive or negative.
- ⋆ Image captioning

  → Find a suitable caption for a picture.
- ⋆ Text generation.

Software

- **Python Version**: Python 3.
- **SciPy**: NumPy, Pandas, and scikit-learn.
- **Keras**: Keras version 2, either a Theano or TensorFlow backend.
- **Jupyter** notebook or **IPython** notebook (not necessary, a text editor + command line or your preferred IDE will do as well).
- Operating System: Windows, Linux or Mac OS X.
- Hardware: A standard modern workstation, no GPUs (graphics processing unit) required.

Open questions

- ⋆ Do these representations correspond in any interpretable way to linguistically motivated representations typically used in theoretical linguistics?
- ⋆ What are the criteria that make one representation better than another?

Source: Younes Samih (2017)

References I

Chollet, F. (2017). *Deep Learning with Python*. Manning Publications.