BSP S3 - Speech Recognition for Luxembourgish by a Recurrent Neuronal Network

Wednesday 27^{th} November, 2019 - 11:24

Le Minh Nguyen
University of Luxembourg
Email: le.nguyen.001@student.uni.lu

Vladimir Despotovic
University of Luxembourg
Email: vladimir.despotovic@uni.lu

***Abstract*—Recurrent Neural Networks (RNN) in comparison with conventional feedforward Neural Networks, use their internal state to operate on data series. One of the domains in which RNNs are applied is speech recognition. In this paper, the focus will be on the classification of spoken words to their according labels. We apply RNN, Long short-term memory (LSTM) and feedforward Neural Networks for the classification and compare their accuracies.**

1. Introduction ($\pm$ 5% of total words)

This paper presents the Bachelor Semester Project (BSP) made by Le Minh Nguyen together with Vladimir Despotovic as his motivated tutor. The project is divided into two periods. In the first period, we implement our classification model to recognize spoken words. The first period contains the following concepts:

- Data preprocessing.
- Training set and testing set.
- Deep Neural Networks architecture: Deep Feedforward Neural Networks and Recurrent Neural Networks, which are presented in the popular Deep Learning textbook [1].

In the second period, we explain the deep Neural Networks architecture and compare their accuracies obtained during our classification experiment.

The length of the report should be from 6000 to 8000 words excluding images and annexes. The sections presenting the technical and scientific deliverables represent $\pm$ 80% of total words of the report.

2. Project description

2.1. Domains

- Speech Recognition
- Artifictial Neural Networks
- Deep Learning
- Data preprocessing
- Training set and testing set
- Python
- Keras

2.1.1. Scientific. The scientific aspects covered by this Bachelor Semester Project are the concepts of speech recognition and Deep Learning. Different Artificial Neural Networks architectures are presented scientifically.

Speech Recognition. The objective of speech recognition is to map an audio signal which contains a set of spoken natural language expressions to the matching sequence of words produced by the speaker.

Artifictial Neural Networks (ANN). ANNs are computing systems inspired by the biological brain. These systems are based on a set of connected units called artificial nodes. Each connection can transmit *signals* between units. A unit can process the signal and transmit it to another unit.

Deep Learning. This field deals with learning by decomposing a task's input into smaller and simpler compositions. With Deep Learning, computing systems can build complex concepts from a composition of simpler concepts.

2.1.2. Technical. The technological aspect which is covered in this project is the data collection, feature extraction and implementation of our classification model.

Data preprocessing. Data preprocessing is an important phase in machine learning. It ensures the quality of the gathered data by eliminating irrelevant and redundant information. Data preprocessing contains tasks such as cleaning, instance selection, normalization, feature extraction and selection. The result of data preprocessing is the training set. We will focus on feature extraction in this paper with the presentation of Mel-frequency cepstral coefficients (MFCC) in section.

Training set and testing set. In order for a computing system to learn from and make predictions on data, a mathematical model is built from an input data. This input data used to create the model consists of two datasets. The training and testing set. The training set contains pairs of an input vector and an output vector often called the label. With the training set, the model learns to map the input vector to the label. Whereas, the testing set evaluates how well the model generalizes the prediction over the dataset.

Python. This is a programming language which is interpreted, high-level and general-purpose. [2]

Keras Library. *Keras* is a high-level Neural Networks API written in Python. It is designed for fast experimentation with Deep Learning. [3]

2.2. Targeted Deliverables

2.2.1. Scientific deliverables. One of the main deliverables is to present the notions of deep learning. This paper should give a small introduction to Neural networks and their application for classification. Further, we extend this scientifical presentation by diving deeper into three different Neural Networks; Feedforward Neural Networks, Recurrent Neural Networks (RNN) and Long short-term memory (LSTM). Additionally, we explain how we extract the features from the dataset. Finally, we compare their accuracies obtained after being trained on the given audio dataset.

Provide a synthetic and abstract description of the scientific deliverables that were targeted to be produced.

2.2.2. Technical deliverables. The other main deliverable for this paper is to implement our classification model based on the three Neural Networks mentioned in the section above. Additionally, we collect a small dataset of Luxembourgish spoken words $w \in \{'0', \ldots, '9', 'moien'\}$ to train our model to classify these words.

Provide a synthetic and abstract description of the technical deliverables that were targeted to be produced.

2.3. Constraints

For this BSP, we set the different constraints for the Neural Networks, for the Luxembourgish vocal dataset and the classification model's implementation.

Classification problem. For the presentation of the three required Neural Network architecture, we set the constraint to present only the Neural Networks in a classification setting rather than for a regression problem.

Data set. First of all, we focus on training our model on a dataset containing English spoken numbers. After having trained the model successfully on this dataset, we train it on the smaller Luxembourgish dataset and analyse its prediction accuracy. Since we aren't able to collect as much voice recordings as the English data set, we should not expect high accuracy.

Framework implementation. We do not implement the Neural Networks architecture since existing deep learning libraries such as Keras are already matured. Thereby, we focus on explaining how the APIs work and how to use them in our model.

Other constraints. Other possible constraints.

3. Pre-requisites ($[5\%..10\%]$ of total words)

Describe in these sections the main scientific and technical knowledge that is required to be known by you before starting the project. Do not describe in details this knowledge but only abstractly. All the content of this section shall not used, even partly, in the deliverable sections.

In order to start on the project, certain skills in programming and mathematics are required. In particular, the preliminary requirement of the project is as follow:

- Understanding of vector and matrix algebra.
- Introductory course in Python.
- Software development.
- Knowledge of probability and statistics, but it is not mandatory.

3.1. Scientific pre-requisites

Linear Algebra.

3.2. Technical pre-requisites

Python.
Software development.

4. A Scientific Deliverable

For each scientific deliverable targeted in section provide a full section with all the subsections described below.

4.1. Requirements ($\pm$ 15% of section's words)

Functional Requirement (FR) and Non-Functional Requirement (NFR)

- **FR01** Present the feature extraction with MFCC
- **FR02** Present the notions of deep learning and Neural Networks
  This should present an introduction to the domain of deep learning. It presents an overview of 3 different Neural Network architectures. Every Neural Network presentation will be divided into a small introduction and theoretical aspect.
- **NFR01** Accuracy comparison
  During this section, we compare and analyse the accuracies obtained from the three Neural Networks architecture.

4.2. Design (± 30% of section's words)

Provide the necessary and most useful explanations on how those deliverables have been produced.

4.2.1. FR01: Deep Learning and Neural Networks.

4.2.2. Feedforward Neural Network.

4.2.3. Recurrent Neural Network.

4.2.4. Long short-term memory.

4.3. Production (± 40% of section's words)

Provide descriptions of the deliverables concrete production. It must present part of the deliverable (e.g. source code extracts, scientific work extracts) to illustrate and explain its actual production.

4.3.1. FR02: Feature extraction with MFCC.

4.3.2. NFR01: Accuracy comparison.

4.4. Assessment (± 15% of section's words)

Provide any objective elements to assess that your deliverables do or do not satisfy the requirements described above.

5. A Technical Deliverable

For each technical deliverable targeted in section provide a full section with all the subsections described below. The cumulative volume of all deliverable sections represents 75% of the paper's volume in words. Volumes below are indicated relative to the section.

5.1. Requirements (± 15% of section's words)

Functional Requirement (FR) and Non-Functional Requirement (NFR)

- **FR01** Implementation of the three classification models
  We use the Keras library to implement our models with three different Neural Network architecture
- **FR02** Collect a small dataset of Luxembourgish spoken words
  We collect a small dataset containing Luxembourgish audio samples containing the words $w \in \{'0', \dots, '9', 'moien'\}$

5.2. Design (± 30% of section's words)

We explain how to use the Keras Library.

5.3. Production (± 40% of section's words)

5.3.1. Implementation of classification models. We present the implementation of our classificatin models

5.3.2. FR02: Luxembourgish dataset collection. We present how we collect our Luxembourgish dataset.

5.4. Assessment (± 15% of section's words)

Acknowledgment

I would like to thank my tutor Vladimir Despotovic for his constructive feedback and mentorship. His introduction and explanation of neural networks were outstanding. I would recommend fellow BiCS Students interested in this field to work with Vladimir Despotovic. Additionally, I thank him for supervising my paper.

6. Conclusion

References

[1] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016, http://www.deeplearningbook.org.

[2] "Python software foundation, python language reference, version 3.7. available at," http://www.python.org/.

[3] F. Chollet *et al.*, "Keras," https://keras.io, 2015.

7. Appendix