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

Tuesday 3rd December, 2019 - 22:46

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

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

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 ± 80% of total words of the report.

***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

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.

2. Project description

2.1. Domains

- Speech Recognition
- Artificial 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. In the past, Automatic Speech Recognition (ASR) was made up of different modules such as complex feature extraction, acoustic models, sequential models, language and pronunciation models. [2] Sequential models, such as Hidden Markov Models (HMM), in combination with a pre-trained language model were used to map sequences of phones to output words. [3] A different approach is to build ASR models end-to-end. With deep learning, it replaces most of the modules with a single module. This alternative method will be the main task of this report.

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 (MFCCs) in section 4.2.1.

Training set and testing set. 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. [4]

Keras Library. *Keras* is a high-level Neural Networks API written in Python. It is designed for fast experimentation with Deep Learning. [5]

2.2. Targeted Deliverables

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

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.

2.3. Constraints

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

Data set. First, we focus on training our model on a dataset containing English spoken numbers. [6] 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.

Speech recognition. Since no datasets with continuous sentences of the Luxembourgish Language exists, we choose to work with isolated word recognition instead of continuous speech recognition. In isolated word recognition, words are separated by pauses.

Speaker dependent. To simplify the collection of the Luxembourgish dataset, we collect the audio recordings from one person which makes the isolated word recognition speaker-dependent.

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.

ANNs implementation. We do not implement the Artificial 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.

3. Pre-requisites

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. is a sub-field of mathematics, which works with vectors and matrices. Since the input of deep learning is data that are transformed into structures of rows and columns, linear algebra is one of the key foundations of deep learning. It is used to describe the operations of the deep learning algorithms, and implement the algorithms in code. A feedforwad ANN can be represented as a composite function which multiplies some matrices and vectors together. All tasks in deep learning relate to linear algebra, from data preprocessing to the deep learning algorithms. [1]

3.2. Technical pre-requisites

Python. Python is an interpreted, high-level and general-purpose programming language, which is conceived in the late 1980s and released in 1991 by Guido van Rossum. [7] Its design philosophy accentuates readability of the code. As many high-level programming languages, Python is dynamically typed. Further, it supports multiple programming paradigms such as procedural, object-oriented and functional programming. I pursued many classes based on Python whether in high school or at university. I have a proficient background in programming in Python to work on this project.

Software development. Software development is the process of designing and implementation of applications and frameworks. In general, the process of software development is writing and maintaining the source code which is often a planned and structured process. The software development contains mostly research, prototyping, modification and reuse of existing software. During the technical deliverable section, we use this structured process to design and implement our ANNs.

4. A Scientific Deliverable

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

- **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 ($\pm$ 30% of section's words)

4.2.1. FR01: Feature extraction with MFCCs.

Before we can train our ASR model, we have to process our audio data. The first task is to extract features from the data which describes natural languages expressions and excludes background noises and emotions.

Mel Frequency Cepstral Coefficients (MFCCs) are a feature generally used in ASR. MFCCs were presented by Davis and Mermelstein in the 1980s. [8]

To extract the MFCCS from the dataset, the following extraction steps have to be executed on a speech signal sampled at $16kHz$:

1. Frame the signal into short frames.

   Frame the signal into $25ms$ frames. The frame length of a $16kHz$ signal:

   0.025*16000 = 400 samples.

   With a frame step of 160 $samples$ or $10ms$, the frames overlap themselves. Such that the first 400 $samples$ frame start at sample 0 and the second frame starts at sample 160. This pattern repeats until the end of speech signal. If the audio file is not divisible by an even number, it is padded by zeros until it is.

2. For each frame calculate the periodogram estimate of the power spectrum. To obtain the Discrete Fourier Transform (DTF) from the frame i, we perform:

   $$S_i(k) = \sum_{N}^{n=1} s_i(n)h(n)e^{-j2\pi kn/N}$$

   For each speech frame $s_i(n)$, the periodogram-based power spectral estimate is calculated with:

   $$P_i(k) = \frac{1}{N}\mid S_i(k) \mid^2$$

   This is called the Periodogram estimate of the power spectrum which identifies for every frame which frequencies are present. The first 257 coefficients are kept from the 512 points fast Fourier transform which is an algorithm to compute the DTF.

3. Apply the Mel filterbank to the power spectral and sum the energy in each filter. The Mel filterbank is a set of 26 triangular filters. The filterbank energies are calculated by multiplying every filterbank with the power spectrum and adding the coefficients kept in the previous step.

4. Take the logarithm of the 26 filterbank energies.

5. Take the Discrete Cosine Transform (DCT) of the 26 log filterbank energies resulting with 26 cepstral coefficients. We only used the lower 12-13 of the 26 coefficients for ASR.

figures to be added

4.2.2. FR02: Deep Learning and Neural Networks.

4.2.3. Feedforward Neural Network.

4.2.4. Recurrent Neural Network.

4.2.5. Long short-term memory.

4.3. Production ($\pm$ 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. Feature extraction with MFCC in Python.

4.4. Assessment ($\pm$ 15% of section's words)

4.4.1. NFR01: Accuracy comparison. 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 ($\pm$ 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 $w \in \{'0', \ldots, '9', 'moien'\}$

5.2. Design ($\pm$ 30% of section's words)

We explain how to use the Keras Library.

5.3. Production ($\pm$ 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 ($\pm$ 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] D. Amodei, R. Anubhai, E. Battenberg, C. Case, J. Casper, B. Catanzaro, J. Chen, M. Chrzanowski, A. Coates, G. Diamos, E. Elsen, J. H. Engel, L. Fan, C. Fougner, T. Han, A. Y. Hannun, B. Jun, P. LeGresley, L. Lin, S. Narang, A. Y. Ng, S. Ozair, R. Prenger, J. Raiman, S. Satheesh, D. Seetapun, S. Sengupta, Y. Wang, Z. Wang, C. Wang, B. Xiao, D. Yogatama, J. Zhan, and Z. Zhu, "Deep speech 2: End-to-end speech recognition in english and mandarin," *CoRR*, vol. abs/1512.02595, 2015. [Online]. Available: http://arxiv.org/abs/1512.02595

[3] W. S. J. Cai, "End-to-end deep neural network for automaticspeech recognition," 2015. [Online]. Available: https://cs224d.stanford.edu/reports/SongWilliam.pdf

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

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

[6] P. Warden, "Speech commands: A dataset for limited-vocabulary speech recognition," *CoRR*, vol. abs/1804.03209, 2018. [Online]. Available: http://arxiv.org/abs/1804.03209

[7] "General python faq," https://docs.python.org/3/faq/general.html#what-is-python, accessed 19/05/19.

[8] "Mel frequency cepstral coefficient (mfcc) tutorial," http://practicalcryptography.com/miscellaneous/machine-learning/guide-mel-frequency-cepstral-coefficients-mfccs/#references, accessed 02/12/19.

7. Appendix