BRAIN2SPEECH - brain signal synthesis

Vanda Réka Halasi
Electrical Engineering and Informatics
BME
Budapest,Hungary
halasivandareka@gmail.com

Zsombor Seres
Electrical Engineering and Informatics
BME
Budapest,Hungary
zsombi9998@gmail.com

Dániel Harsányi
Electrical Engineering and Informatics
BME
Budapest,Hungary
danielharsanyi18@gmail.com

***Abstract*—Brain-computer interfaces (BCIs) are applications that can translate the user's neural activity into computer-interpretable instructions, bridging the gap between the user's central nervous system and the environment. There are numerous applications for BCIs, one of the most important of which is in medicine, where, in addition to detecting neurological diseases, they can be used to control prosthetic limbs, wheelchairs or even talking devices with brain signals.**

If we could synthesise speech from the recorded signals of our BCI-s we could help people who lost their ability to speak and it would bring us closer to understanding speech production. In our work we have tried to do this by using deep learning methods.

***Index Terms*—DeepLearning, AI, neuralnets, LSTM, brain-signs**

I. Introduction

Speech production is a very complex process involving a large number of muscles and cognitive processes which is not yet fully understood. In recent years, research and development of BCI applications has made outstanding progress but the understanding of neural activity during speech could not develop at the same pace. This is because speech is a uniquely human ability so it can not be investigated with animal experiments and using invasive BCIs (the ones with the best spatial signal resolution) on humans have legal limits.

Deep learning methods are often used for solving problems that cannot be described analytically. It is especially useful in this case because of the complexity of the problem. We used a dataset containing timeseries and stimulus data to directly convert brain signals to speech with end-to-end training.

II. Motivation

A. Previous solutions

Despite the fact that a full understanding of speech production is currently lacking, research of BCIs that offer speech neuroprosthesis have recently gained popularity [3], [4]. It is proven that decoding of a textual representation is possible from neural recordings during actual speech production using phonemes, words and even full sentences [5]–[7]. Some studies aimed at directly synthesizing an audio waveform from the neural data to discover more natural connections [8]. Some recent advencements show that the decoding is possible from imagined speech as well [9].

Most of these studies use electrocorticography (ECoG), an invasive recording modality of neural activity that provides high temporal and spatial resolution and high signal-to-noise ratio [10]. An alternative measurement type is stereotactic EEG (sEEG), where electrode shafts are implanted into the brain through small burr holes. It is less invasive than ECoG and instead of high density coverage of specific regions, it provides sparse sampling of multiple regions.

B. Theoretical foundations

1) SingleWordProductionDutch Dataset [2]: This dataset provides data of 10 Dutch participants speaking prompted words aloud while audio and intracranial EEG data are recorded simultaneously.

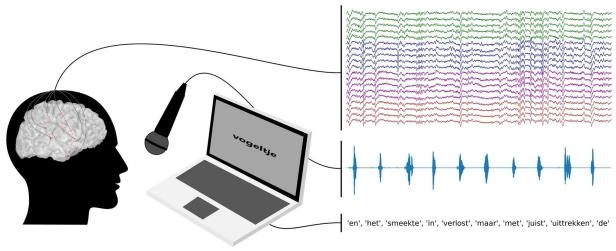

Fig. 1. Measurement

Participants were asked to read aloud words that were shown to them on a laptop screen. One random word was presented on the screen for a duration of 2 seconds during which the participant read the word aloud once. After the word, a fixation cross was displayed for 1 second. This was repeatedfor a total of 100 words, resulting in a total recording time of 300 seconds for each participant.

The dataset is advantageous for this task because sEEG electrodes were used for the recording so multiple brain areas were recorded simultaneously, leveraging multiple sources of potential information. On average, there were 110.3 working electrodes implanted per 1 participant. To take advantage of this feature, exact electrode locations were detected using pre-implantation Magnetic Resonance Imaging (MRI) and post-implantation Computer Tomography (CT). An anatomical location label was extracted from the Destrieux atlas [11] based parcellation for each electrode.

The raw data files (XDF format) were converted to Neurodata Without Borders (NWB - https://www.nwb.org/) format and organised in the iBIDS [12] data structure format using Python scripts. The creators of the dataset also provided useful preprocessing functions for us to use.

2) LSTM - Long-Short Term Memory [1]: The LSTM model was created in 1997 to develop recurrent neural network's memory with the ability of remembering long-term relations. It is built up by three different kind of gates, such as forget, input and output. One of the key elements of the architecture is the cell state, which can be imagined as a car which goes to the highway memorizing important and forgetting useless information. The gates and cell state's mathematical description is shown in the below equations.

$$f_t = \sigma(W_{xf}x_t + W_{hf}h_{t-1} + W_{cf} \odot c_{t-1} + b_f) \quad (1)$$

$$i_t = \sigma(W_{xi}x_t + W_{hi}h_{t-1} + W_{ci} \odot c_{t-1} + b_i) \quad (2)$$

$$c_t = f_t \odot c_{t-1} + i_t \odot tanh(W_{xc}X_t + W_{hc}h_{t-1} + b_c) \quad (3)$$

$$o_t = \sigma(W_{xo}x_t + W_{ho}h_{t-1} + W_{co} \odot c_{t-1} + b_o) \quad (4)$$

These equations are in the Vanilla Transformer model where every gate uses sigmoid function. The `c` variable stands for the cell state, `W` is the weight, `b` is bias and `f, i, o` are the gates in order.

$$h_t = o_t \odot tanh(c_t) \quad (5)$$

The upper equation defines the hidden state of the model in every step.

Our LSTM modell is used only for predicting one sequence and a linear projector is used to provide the output sequence.

III. Our solution

IV. Implementation

A. Technology

Google Colaboratory was used to run the trainings, because it is one of the easiest way to gain access to GPUs. The LSTM model is not the most quickest neural network and we used a huge amount of data. The solution for us was to create a shared drive folder, where data and the result of the running are stored. The other advantage was that everyone had access to it from everywhere in a simple way. We mostly used `Pytorch` to create datasets, dataloaders and the model also.

B. Preprocessing

We used most of the preprocessing steps such as data cleaning and filtering made by the creators of the SingleWordProductionDutch Dataset. Feature extraction of iEEg Data (applied to all electrode channels):

1) Bandpass filtering 70-170 Hz to extract High-Gamma Band (IIR filter, order 4)
2) Bandstop filtering 98-102 Hz to attenuate first harmonic of line noise (IIR filter, order 4)
3) Bandstop filtering 148-152 Hz to attenuate first harmonic of line noise (IIR filter, order 4)
4) Hilbert transform to get the envelope of the signal
5) Averageing in frequency domain with 50 ms window and 10 ms frameshift
6) Stacking non-overlapping neighboring windows up to 200 ms to include temporal information

Feature extraction of audio Data:

1) Downsampling to 16 kHz
2) Applying short-term-fourier-transform in windows of 50 ms with a frameshift of 10 ms to get the spectogram
3) Converting spectogram to log-mel representation

These steps were implemented in Python, using the scipy package.

C. Dataloaders

There are recordings of 10 participants and we give the user the possibility to choose the participants, by giving a list of their numbers to the data extractor script called `get_data`. Using the features and spectograms extracted by the helper functions a Pytorch dataset classes were created to normalize the data and it can load different size of input and output sequences. The Dataset gives back the previously filtered and selected channels as an input the spectogram itself for output.

Three Pytorch Dataloaders are created for train, validation and test splits with a given batch size. Train data is loaded shuffled, unlike test and validation sets.

D. Training

Our training loop is implemented in a notebook which is running in colab. It uses a data set saved in a Google drive folder and saves every loss into a folder from where it can be easily visualize with `tensorboard`. An example can be seen in 2 image.

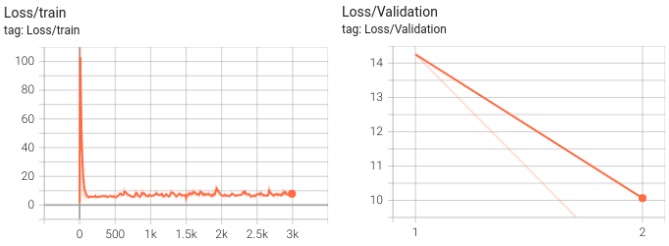

Fig. 2. Losses while training

Our hiperparameter optimization was quite tough process because the feature set is large and 16000 sampling/sec amount is not easy to learn. We tried out really few combinations because of the colaboraty limitations, we could only train it 3 or 4 times. A visualization also made from the reconstructed audio files and it not really shows any correlation to the target data. The reason is that out model is not complex enough for the data and it can't really handle the complexity of it.

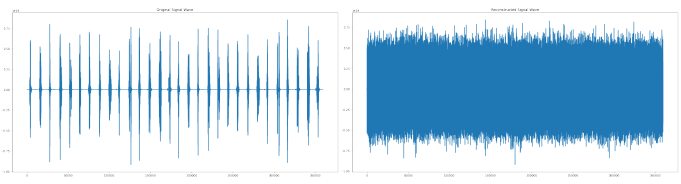

Fig. 3. Real (on the left) and reconstrcuted (on the right) waves

In the prevoius section we mentioned the visualization of the audio files, which can be seen on the 3 image.

REFERENCES

[1] Sepp Hochreiter – Jürgen Schmidhuber: Long short-term memory. Neural computation, 1997, 1735–1780. p.

[2] Christian Herff, Maxime Verwoert: Dataset of Speech Production in intracranial Electroencephalography, 2022, DOI 10.17605/OSF.IO/NRGX6.

[3] Rabbani, Q., Milsap, G. and Crone, N.E. The Potential for a Speech Brain–Computer Interface Using Chronic Electrocorticography. Neurotherapeutics 16, 144–165 (2019). https://doi.org/10.1007/s13311-018-00692-2

[4] Moses, D. A. et al. Neuroprosthesis for decoding speech in a paralyzed person with anarthria. New England Journal of Medicine 385, 217–227 (2021).

[5] Ramsey, N. F. et al. Decoding spoken phonemes from sensorimotor cortex with high-density ECoG grids. Neuroimage 180, 301–311 (2018).

[6] Kellis, S. et al. Decoding spoken words using local field potentials recorded from the cortical surface. Journal of Neural Engineering 7 (2010).

[7] Makin, J. G., Moses, D. A. and Chang, E. F. Machine translation of cortical activity to text with an encoder–decoder framework. Tech. Rep., Nature Publishing Group (2020).

[8] Anumanchipalli, G. K., Chartier, J. and Chang, E. F. Speech synthesis from neural decoding of spoken sentences. Nature 568, 493–498, https://doi.org/10.1038/s41586-019-1119-1 (2019).

[9] Proix, T. et al. Imagined speech can be decoded from low-and cross-frequency intracranial EEG features. Nature communications 13, 1–14 (2022).

[10] Parvizi, J. and Kastner, S. Promises and limitations of human intracranial electroencephalography. Nature neuroscience 21, 474–483 (2018).

[11] Destrieux, C., Fischl, B., Dale, A. and Halgren, E. Automatic parcellation of human cortical gyri and sulci using standard anatomical nomenclature. NeuroImage 53, 1–15, https://doi.org/10.1016/j.neuroimage.2010.06.010 (2010).

[12] Holdgraf, C. et al. iEEG-BIDS, extending the brain imaging data structure specification to human intracranial electrophysiology. Scientific data 6, 1–6 (2019).