Modeling brain responses to perceived speech with LSTM networks

Julia Berezutskaya JU.BEREZUTSKAYA@GMAIL.COM
Zachary V. Freudenburg Z.V.FREUDENBURG@UMCUTRECHT.NL
Nick F. Ramsey N.F.RAMSEY@UMCUTRECHT.NL
Brain Center Rudolf Magnus, Department of Neurology and Neurosurgery, University Medical Center Utrecht, Heidelberglaan 100, 3584 CX, Utrecht, The Netherlands

Umut Güçlü U.GUCLU@DONDERS.RU.NL
Marcel A.J. van Gerven M.VANGERVEN@DONDERS.RU.NL
Radboud University, Donders Institute for Brain, Cognition and Behaviour, Montessorilaan 3, 6525 HR, Nijmegen, The Netherlands

Keywords: LSTM, RNN, brain responses, speech

Abstract

We used recurrent neural networks with long-short term memory units (LSTM) to model the brain responses to speech based on the speech audio features. We compared the performance of the LSTM models to the performance of the linear ridge regression model and found the LSTM models to be more robust for predicting brain responses across different feature sets.

1. Introduction

One of the approaches to understanding how the human brain processes information is through modeling the observed neural activity evoked during an experimental task. Typically, the neural activation data are collected as a response to a set of stimuli, for example pictures, audio or video clips. Then, salient features are extracted from the stimulus set and used to model the neural responses. The learned mapping is called a neural encoding model (Kay et al., 2008; Naselaris et al., 2012).

A common approach is to use hand-engineered features, which can be complex transformations of the stimulus set and learn a linear mapping between the stimulus features and the neural responses. In case of speech, the spectrogram and non-linear spectrotemporal modulation features have been used in linear encoding models (Santoro et al., 2014).

Non-linear models of neural encoding have recently started to gain popularity in the neuroscience community, since they allow learning a more complex mapping between the stimulus features and the neural responses. In a recent study, various models from the recurrent neural network family were trained to predict the neural responses to video clips (Güçlü & van Gerven, 2017).

In the present study, we apply LSTM models to predict the neural responses to continuous speech. We use various sets of stimulus features for model training and compare the performance of the LSTM models with the performance of a linear encoding model.

2. Methods

Brain data collection and preprocessing

Fifteen patients with medication-resistant epilepsy underwent implantation of subdural electrodes (electrocorticography, ECoG). All patients gave written consent to participate in research tasks alongside the clinical procedures to determine the source of the epileptic activity. During the research task, the patients watched a 6.5 min short movie with a coherent plot (fragments of Pippi Longstocking, 1969) while their neural activity was recorded through the ECoG electrodes. The ECoG recordings were acquired with a 128 channel recording system (Micromed, Treviso, Italy) at a sampling rate (SR) of 512 Hz filtered at 0.15-134.4 Hz. All patients had electrodes in temporal and frontal cortices, implicated in auditory and language

Preliminary work. Under review for Benelearn 2017. Do not distribute.

processing (Howard et al., 2000; Hickok & Poeppel, 2007; Friederici, 2012; Kubanek et al., 2013).

The collected ECoG data were preprocessed prior to model fitting. Per patient, based on the visual inspection, electrodes with noisy or flat signal were excluded from the dataset. Notch filter at 50 and 100 Hz was used to remove line noise and common average re-referencing was applied. The Gabor wavelet decomposition was used to extract neural responses in the high frequency band (HFB, 60-120 Hz) from the time domain signal. The Wavelet decomposition was applied in the HFB range in 1 Hz bins with decreasing window length (4 wavelength full-width at half max). The resulting signal was averaged over the whole range to produce a single HFB neural response per electrode. The resulting neural responses were downsampled to 125 Hz. The preprocessed data were concatenated across patients over the electrode dimension (total number of electrodes = 1283).

Audio features

The soundtrack of the movie contained speech and music fragments. From the soundtrack, we constructed three input feature sets for training the models. First, we extracted the waveform of the movie soundtrack and downsampled it to 16000 Hz. To create the first, time-domain, feature set (*time*), we reshaped the waveform to the matrix of size $N \times F_1$, where N is the number of time points at the SR of the neural responses (125 Hz), and F_1 is 128 time features (16000/125) To make the second feature set, we extracted a sound spectrogram at 128 logarithmically spaced bins in range 180-7000 Hz. This resulted in a $N \times F_2$ matrix with $F_2 = 128$ features (*freq*). Finally, the spectrogram was filtered with a bank of 2D Gabor filters to extract spectrotemporal modulation energy features (Chi et al., 2005). The filtering was done at 16 logarithmically spaced bins in range 0.25-40 Hz along the temporal dimension, and 8 logarithmically spaced bins in range 0.25-4 cyc/oct along the frequency dimension. The third feature matrix $N \times F_3$ was built by concatenating all spectrotemporal modulation features: 16×8, $F_3 = 128$ features (*smtm*). The spectrogram and the spectrotemporal modulation energy features were obtained using the NSL toolbox (Chi et al., 2005).

Linear encoding model

For each input feature set, a separate kernel linear ridge regression (Murphy, 2012) was trained to predict the neural responses to speech fragments. The HFB neural response of each electrode y_e at time point t was modeled as a linear combination of the input audio features at this time point:

$$y_e(t) = \boldsymbol{\beta}_e{}^\top \mathbf{x}(t) + \epsilon_e$$

where $\epsilon_e \sim \mathcal{N}(0, \sigma^2)$.

L^2 penalized least squares loss function was analytically minimized to estimate the regression coefficients $\boldsymbol{\beta}_e$. The kernel trick was used to avoid large matrix inversions in the input feature space:

$$\boldsymbol{\beta}_e = \mathbf{X}^\top (\mathbf{X}\mathbf{X}^\top + \lambda_e \mathbf{I}_n)^{-1} \mathbf{y}_e$$

where n is the number of training time points.

A nested cross-validation was used to estimate the amount of regularization λ_e (Güçlü & van Gerven, 2014). First, a grid of the effective degrees of freedom of the model fit was specified. Then, Newton's method was used to solve the effective degrees of freedom for λ_e. Finally, λ_e that resulted in the lowest nested cross-validation error was taken as the final estimate.

The model was tested on 5% of all data. A five-fold cross-validation was used to validate the model performance. In each cross-validation fold different speech fragments were selected for testing, so that no data points were shared in test sets across five folds.

Model performance was measured as the Spearman correlation between predicted and observed neural responses in the test set. The correlation values were averaged across five cross-validation folds and were transformed to t-values for determining significance (Kendall & Stuart, 1961).

LSTM encoding models

For each input feature set, six LSTM models (Hochreiter & Schmidhuber, 1997) with varying architectures were trained to predict the neural responses to speech fragments. The six LSTM models were specified using a varying number of hidden layers (one or two) and a varying number of units per hidden layer (20, 50 or 100). A fully-connected linear layer was specified as the output layer. The neural response of each electrode y_e at time point t was modeled as a linear combination of the hidden states $\mathbf{h}(t)$. For models with one hidden LSTM layer (*1-lstm20, 1-lstm50, 1-lstm100*):

$$y_e(t) = \boldsymbol{\beta}_e^\top \mathbf{h_1}(t) + b_e + \epsilon_e$$

where b_e is a bias and $\epsilon_e \sim \mathcal{N}(0, \sigma^2)$.

For models with two hidden LSTM layers (*2-lstm20, 2-lstm50, 2-lstm100*):

$$y_e(t) = \boldsymbol{\beta}_e^\top \mathbf{h}_2(t) + b_e + \epsilon_e$$

The hidden states $\mathbf{h}_1(t)$ were computed in the following way:

$$\begin{aligned}
\mathbf{f}(t) &= \sigma\left(\mathbf{U}_f\mathbf{h}_1(t-1) + \mathbf{W}_f\mathbf{x}(t) + \mathbf{b}_f\right) \\
\mathbf{i}(t) &= \sigma\left(\mathbf{U}_i\mathbf{h}_1(t-1) + \mathbf{W}_i\mathbf{x}(t) + \mathbf{b}_i\right) \\
\mathbf{o}(t) &= \sigma\left(\mathbf{U}_o\mathbf{h}_1(t-1) + \mathbf{W}_o\mathbf{x}(t) + \mathbf{b}_o\right) \\
\mathbf{c}(t) &= \mathbf{i}(t)\tanh\left(\mathbf{U}_c\mathbf{h}_1(t-1) + \mathbf{W}_c\mathbf{x}(t) + \mathbf{b}_c\right) \\
&\quad + \mathbf{f}(t)\mathbf{c}(t-1) \\
\mathbf{h}_1(t) &= \mathbf{o}(t)\tanh\left(\mathbf{c}(t)\right)
\end{aligned}$$

where σ is the logistic sigmoid function. Vectors $\mathbf{f}(t)$, $\mathbf{i}(t)$, $\mathbf{o}(t)$ and $\mathbf{c}(t)$ correspond to four LSTM gates: *forget gate*, *input gate*, *output gate* and *cell state*, respectively. Matrices $\mathbf{U}$ and $\mathbf{W}$ contain the gate-specific weights and vectors $\mathbf{b}$ are the gate-specific bias vectors.

For models with two hidden LSTM layers (*2-lstm20, 2-lstm50, 2-lstm100*), the hidden states $\mathbf{h}_2(t)$ were computed in the similar way, except that the input to the cells at the second layer was $\mathbf{h}_1(t)$.

Mean squared error function was minimized during the training using Adam optimizer (Kingma & Ba, 2014). The models were trained using backpropagation through time, with truncation after the i-th time point, corresponding to the 500 ms lag. Each model was optimized using a validation set (5% of all data) and early stopping: training was stopped if the loss on the validation set did not decrease for 30 epochs. The model with least loss on validation set was used as the final model. Each model was tested on 5% of all data. Chainer package (Tokui et al., 2015) was used for implementing the LSTM models.

The model performance was computed in the same way as for the linear model. Similarly, a five-fold cross-validation was used to validate the model performance. The correlations were transformed to t-values for determining significance.

Model performance comparison

For each model, the model performance scores (Spearman correlations) were thresholded at $p < .001$, Bonferroni corrected for the number of electrodes. Per each feature set, we selected the electrodes with significant performance across all the models: 53 electrodes for *freq*, 125 electrodes for *smtm* and 0 for *time*. The performance across the models was compared using one-way ANOVA test. Tuckey's honest significant difference (HSD) test was used post-hoc to determine pairs of models with significantly different mean performance values. Separately, per each model, we calculated the number of electrodes for which the model achieved significant performance.

3. Results

When trained on *time*, the performance of the linear ridge regression model was not significant $p < .001$, Bonferroni corrected. All the LSTM models performed significantly above chance. When trained on *freq*, there was significant difference in performance between the linear ridge regression model and the LSTM models ($F(1119) = 12.65, p = 5.69 \times 10^{-13}$, Tuckey's HSD test: each pair of ridge−LSTM means were significantly different at $p = 0.001$). When trained on *smtm*, there was no significant difference between the linear ridge regression model and the LSTM models ($F(874) = 1.7, p = .12$). Overall, the LSTM models showed good performance with all three feature sets (Fig. 1). The performance of the linear ridge regression model depended strongly on the input feature set and improved as the input features became more complex. Despite varying the parameters of the LSTM architecture, there was almost no difference in performance among the six LSTM models. We observed a significant difference in LSTM model performance for the *time* feature set: $F(275) = 9.37, p = 2.99 \times 10^{-8}$. For both one- and two-layer LSTMs, the models with 20 hidden units performed worse compared to the models with a larger number of hidden units (based on the HSD test).

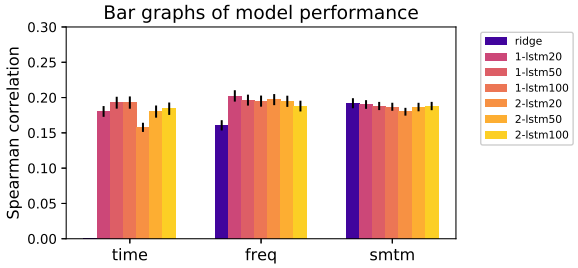

Figure 1. Model performance comparison between the linear ridge regression model and the six LSTM models, trained on separate feature sets: *time*, *freq* and *smtm*. The bars show mean model performance scores over the electrodes (Spearman correlations). The scores were significant at $p < .001$, Bonferroni corrected for the number of electrodes. Error bars indicate standard error of the mean.

All models trained on all feature sets performed significantly above chance only in a subset of all electrodes. For all feature sets, the LSTM models achieved significant performance in a larger amount of electrodes, compared to the linear ridge regression model (Table 1).

All models but the linear ridge regression model

Table 1. Percentage of electrodes the models performed well for when trained on each feature set. Total number of electrodes (100%) is 1283. Highest values are in bold.

MODEL	TIME	FREQ	SMTM
RIDGE	0%	6%	16%
1-LSTM20	**8**%	10%	19%
1-LSTM50	**9**%	**12**%	**25**%
1-LSTM100	7%	**12**%	**28**%
2-LSTM20	7%	11%	17%
2-LSTM50	5%	**12**%	23%
2-LSTM100	**8**%	**12**%	**26**%

trained on *time*, showed significant performance in electrodes located in the temporal cortex (superior temporal gyrus), implicated in auditory processing (Howard et al., 2000; Norman-Haignere et al., 2015). The LSTM models trained on *time* performed significantly well for the electrodes in the superior temporal gyrus. The LSTM models trained on *freq* and *smtm* showed involvement of the electrodes located in the frontal cortex, as well as the posterior middle temporal and parietal cortices (Fig. 2). These cortical regions are implicated in language and other higher-level cognitive processing (Hagoort, 2013; Friederici, 2012).

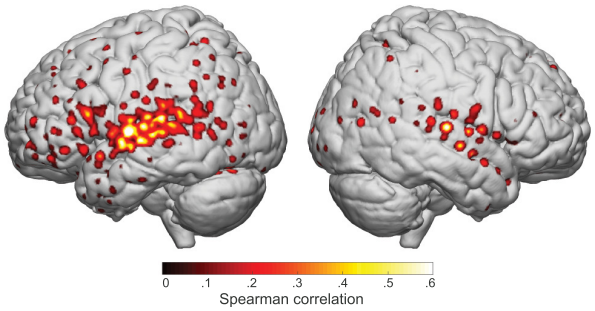

Figure 2. Cortical locations of the electrodes whose responses were modeled significantly above chance (at $p < .001$, Bonferroni corrected for the number of electrodes) by 1-LSTM50 trained on *smtm* feature set.

4. Discussion

In the present study we trained several models to predict the neural responses to perceived speech. The neural responses were obtained using ECoG. We considered a linear ridge regression model and recurrent neural network models with LSTM units varying in architecture. Each model was trained on three separate sets of audio features. We found that the performance of the linear ridge regression model depended strongly on the set of the input features. Notably, the linear ridge regression model did non achieve significant performance using the time domain features. In contrast, the LSTM models showed comparable performance across different feature sets. Using more complex audio features allowed the LSTM models to make accurate predictions for a larger set of ECoG electrodes.

There are multiple reasons why the linear ridge regression model and the LSTM models might have shown different performance when trained on *time* and *freq*. For example, the linear ridge regression model was regularized as opposed to the LSTM models presented here. Additionally, we retrained the LSTM models using a weight decay parameter for regularizing the network weights. The amount of the weight decay was cross-validated, but in multiple cases its optimal value turned out to be zero, and the overall performance of the LSTM models did not change considerably.

Other factors contributing to the superior performance of the LSTM models include presence of non-linear transformations within the LSTM cells (σ and tanh), as well as the *cell states* $\mathbf{c}$ which accumulate the information relevant for the predictions over time. Finally, the linear ridge regression model and the LSTM models differed considerably with respect to the number of the free parameters. We found it challenging to match the linear regression and neural network models with respect to all mentioned issues. Further work is necessary to determine which concrete properties of the LSTM models allowed it to outperform the linear ridge regression model when trained on *time* and *freq*.

The present work has a number of limitations. Because the placement of the ECoG grids varies across patients (depending on the tentative source of epilepsy), it is usually challenging to generalize the model performance to new patients' data. Here we used data from all patients to train the models. The model performance was then cross-validated using a five-fold cross-validation. Increasing the amount of patients could provide a larger overlap in the location of the electrodes. Then, a generalization to the data of unseen patients could be attempted.

5. Conclusions and future work

We trained several LSTM models to predict neural responses to speech based on the speech audio features and compared it to the performance of the linear ridge regression model. In general, the performance of the LSTM models was superior to the performance of the

linear ridge regression model in terms of the prediction accuracy and the amount of electrodes the models were successfully fit for. Further work is planned to investigate in detail what factors contribute to the superior performance of the LSTM models, compared to the linear ridge regression model. Some work on exploring the internal representations learned by the LSTM models (*cell states*) is also planned. Finally, we intend to compare the performance of RNNs with the performance of a convolutional neural network, trained on the wavelet-decomposed audio signal to predict the brain responses.

Acknowledgments

The work was supported by the NWO Gravitation grant 024.001.006.

References

Chi, T., Ru, P., & Shamma, S. A. (2005). Multiresolution spectrotemporal analysis of complex sounds. *The Journal of the Acoustical Society of America*, *118*, 887–906.

Friederici, A. D. (2012). The cortical language circuit: from auditory perception to sentence comprehension. *Trends in cognitive sciences*, *16*, 262–268.

Güçlü, U., & van Gerven, M. A. (2014). Unsupervised feature learning improves prediction of human brain activity in response to natural images. *PLoS Comput Biol*, *10*, e1003724.

Güçlü, U., & van Gerven, M. A. (2017). Modeling the dynamics of human brain activity with recurrent neural networks. *Frontiers in Computational Neuroscience*, *11*, 10–3389.

Hagoort, P. (2013). Muc (memory, unification, control) and beyond. *Frontiers in Psychology*, *4*, 416.

Hickok, G., & Poeppel, D. (2007). The cortical organization of speech processing. *Nature Reviews Neuroscience*, *8*, 393–402.

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, *9*, 1735–1780.

Howard, M. A., Volkov, I., Mirsky, R., Garell, P., Noh, M., Granner, M., Damasio, H., Steinschneider, M., Reale, R., Hind, J., et al. (2000). Auditory cortex on the human posterior superior temporal gyrus. *Journal of Comparative Neurology*, *416*, 79–92.

Kay, K. N., Naselaris, T., Prenger, R. J., & Gallant, J. L. (2008). Identifying natural images from human brain activity. *Nature*, *452*, 352–355.

Kendall, M. G., & Stuart, A. (1961). The advanced theory of statistics (vol. 2), london: Charles w. *Griffin and Co., Ltd*, 1959–1963.

Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. *CoRR*, *abs/1412.6980*.

Kubanek, J., Brunner, P., Gunduz, A., Poeppel, D., & Schalk, G. (2013). The tracking of speech envelope in the human cortex. *PloS one*, *8*, e53398.

Murphy, K. P. (2012). *Machine learning: a probabilistic perspective*. MIT press.

Naselaris, T., Stansbury, D. E., & Gallant, J. L. (2012). Cortical representation of animate and inanimate objects in complex natural scenes. *Journal of Physiology-Paris*, *106*, 239–249.

Norman-Haignere, S., Kanwisher, N. G., & McDermott, J. H. (2015). Distinct cortical pathways for music and speech revealed by hypothesis-free voxel decomposition. *Neuron*, *88*, 1281–1296.

Santoro, R., Moerel, M., De Martino, F., Goebel, R., Ugurbil, K., Yacoub, E., & Formisano, E. (2014). Encoding of natural sounds at multiple spectral and temporal resolutions in the human auditory cortex. *PLoS Comput Biol*, *10*, e1003412.

Tokui, S., Oono, K., Hido, S., & Clayton, J. (2015). Chainer: a next-generation open source framework for deep learning. *Proceedings of Workshop on Machine Learning Systems (LearningSys) in The Twenty-ninth Annual Conference on Neural Information Processing Systems (NIPS)*.