Predicting Brain Image Activation with Long Short-Term Neural Networks

Aditri Bhagirath
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA
abhagira@andrew.cmu.edu

Advised by:
Leila Wehbe and Mariya Toneva
Machine Learning Department, School of Computer Science
Carnegie Mellon University
Pittsburgh, PA
lwehbe@andrew.cmu.edu, mtoneva@andrew.cmu.edu

Introduction

CMU's Brain Image Analysis Research group, and its efforts to gauge how language is represented in the brain, motivated this study. Here, we focus primarily on textual input and its corresponding effects on brain activation on various subjects. Determining which areas of the brain are activated upon exposure to text can provide valuable insight into brain function, and help understand whether activation is more likely to be localized to certain areas. This is immensely applicable when considering the development of BCIs (Brain Computer Interfaces), which are devices allowing direct communication between the brain and an external device. Understanding how language affects brain activation is central to the development of BCIs, which in turn have numerous applications in the fields of medicine, specifically in the improvement of quality of life for disabled individuals.

In this study, we're attempting to assess whether future states of brain activation can be predicted based solely on past states of activation, using LSTMs (Long Short-Term Neural Networks). This architecture was deemed suitable

since LSTMs have the ability to process related data sequences, and are typically used for applications such as connected handwriting recognition or predictive text. LSTMs have been widely used to preserve state and make textual or time series predictions, and we wanted to extend this idea to brain image data, since it seems likely that past areas of activation are likely to affect future areas of activation. While LSTMs have widely been used on textual input, their use on image data is quite uncommon currently.

The Brain Image Analysis Research Group at CMU has made several contributions in the areas of language mapping in the brain, specifically gauging areas of the brain that are activated in response to different kinds of semantics. However, these previous studies focus primarily on text and its effects on the brain. Our study focuses solely on brain activation and its ability to predict future activation in the context of language exposure, aiming to gauge whether past brain state is a reliable indicator of the future.

The data we used consists of MEG (Magnetoencephalography) brain scans from research subjects as they read excerpts from *Harry Potter and the Sorcerer's Stone*. Several different LSTM models were tested and considered, and were used to predict various time-series of brain activation, with varying numbers of steps in the time series. This helps to determine the nature of propagation of predictions of brain activations in the future. More specifically, it allows us to gauge whether accuracy in predictions is preserved as the number of time steps increases, providing insight into the mechanisms of how long brain activations persist and affect future activations. Significant levels of agreement between real and predicted activation were found for some values of time steps.

Additionally, magnitudes of activation in the brain were found to be very similar across words, for corresponding timelines in word exposure. This suggests similarities in the way our brains react to textual data, and reveals patterns in the progression of brain activation from initial exposure to following time periods for each of the words included in the dataset.

Transformation of Sensor Data and Formation of Training and Test Sets

The dataset used for this study is in the form of a 5176 X 20 X 306 matrix. 5176 corresponds to the number of words in the except that participants were made to read. The array entry corresponding to each word is itself a 20 X 306 matrix. Since each word was presented for a period of 500 milliseconds, and MEG scans were taken at 25 millisecond intervals, the entirety of the data for any given word comprises of 20 scans. For each scan, the data was encoded as a vector of 306 values representing the 306 different sensors of the MEG scan, each of which corresponds to a different region in the brain. The data was transformed into the 2-dimensional form below for the purposes of training and prediction, so that we could get a contiguous time series of duration (5176*500) milliseconds, i.e. 43 minutes. The new dimensions of this dataset are 103520 (5176 * 20) X 306, and each point in the dataset corresponds to a single brain scan, represented by a vector of dimensions 1 X 306. The first 81900 of these scans were used for training, while the rest of the values comprised the test set and was used to determine model accuracy.

Sensors →

Time ↓	S-0	S-1	S-2	S-3	…….	S-305
T1=0						
T2=25						
T3=50						
T4=75						
T5=100						
…….						

Figure 1: Data reshaped as a list of 5176 X 20 samples. The first 81900 points used for training the model, the rest is test data

Initial Model

For the first LSTM model developed, the unmodified data set was used as training data. Thus, a single training point consisted of a two-dimensional matrix of dimensions 20 X 306. For prediction, each of the 1176 test points (also consisting of 20 X 306 dimension matrices) was used to predict the next time step, with each output having the same dimensions as this input. Predicted matrices were compared with actual matrices, and their correlation was calculated by computing the z-score of each matrix.

In order to determine how to construct a suitable LSTM model, various parameters had to be taken into consideration. This includes the choice of the number of layers to include, the loss function to use, the optimizer, and the number of hidden units to use per hidden layer. Python's keras and sklearn libraries are ideally suited to developing models for deep learning, and were used in the construction of these models. Parameter tuning was performed by

evaluating performance on the test set by determining correlation values between real values and predicted values of vectors, and also by assessing how the value of the chosen loss function decreased and converged after successive "epochs" or iterations of training.

The LSTM model chosen consists of one input layer, one output layer, and three hidden layers. Each of the hidden layers has 50 hidden units. The loss function used here is the mean squared error, and the "adam" optimizer was used. A batch size of 32 was used for training, with 100 epochs. This first model is stateless, which means that once trained and used to make predictions, the model does not update its state to reflect previous predictions.

Final Stateful Model

Since it seems likely that previous brain activations affect future activations, in order to predict a time-series of activations, the initial LSTM model was modified to preserve state. We did this so that, once the model has been fully trained and is asked to make predictions, it's prediction for time step two will, for instance, take into account its prediction for point 1. This also holds for other future predictions. This change was made because more insightful trends are likely to be determined when given added context. The training data used for this model resembled that in Figure 1, and differed from the initial model in that each training point was a 1 X 306 dimensional vector representing a single scan, instead of a 20 X 306 block. This allowed us to look at predictions at a finer granularity, considering individual scans instead of blocks of scans. It also enabled us to predict time-series of scans, for various values of n (where n represents the length of the time series). Predicting time series and assessing

how strong the correlation is compared to real time series for the test data can additionally help provide insight into how long we can preserve accuracy of predictions. This in turn can enable us to determine how well past activations are likely to influence current activations.

Additionally, the 1 X 306 prediction provides the added advantage of ease of visualization. Even though mean correlation can be computed for 20 X 306 scan blocks, it is difficult to visualize this high-dimensional data as a graph. 1 X 306 dimension predictions allow us to average correlations across sensors for each of the 306 sensor values, and these averages can easily be plotted, like in the graphs shown below. Additionally, a visualization mechanism was created using the MNE module for Python, which is an open-source software for visualizing MEG data and the magnitudes of activation at different areas of the brain (since each sensor value corresponds to a particular location, and the predicted activation vector can simply be passed into the visualizer).

Experimental Setup And Evaluation

The first set of experiments involved predicting time-series of length 1, 2, and 3. This was done using the following function predictWithStateful (subset, startI, prevSteps, numIter, endI). The arguments to this function are specified as follows:

subset → The set of test data that we want to use to make predictions.

startI → The index of the test data, from which we want to start predicting time series. The default value for this is 100, because the first 100 observations are used to configure the LSTM's internal state in order for subsequent predictions to reflect observed trends.

prevSteps → Reflects the number of previous predictions we want to make before generating the prediction for the current time point. This is done because the LSTM model used here is stateful, and we configure the LSTM's internal state for each prediction to add activation context.

numIter → The length of the time series that we want to predict. For instance, if numIter = 2, we predict the first point from the real brain scan value, then predict the second point from the prediction of the first point. So assuming the input is point 100, we're generating point 102.

endI → Indicates the last index from which we want to create a time series prediction.

```
def predictWithStateful(subset, startI, prevSteps, numIter, endI):
    all = []
    for i in range(startI, endI):
        for j in range(i-prevSteps, i):
            current = subset[j]
            shapedCurrent = np.reshape(current, (1, 1, 306))
            loaded_model.predict(shapedCurrent)
        stept = subset[i]
        shapedStept = np.reshape(stept, (1, 1, 306))
        for k in range(numIter):
            stepTPlus1 = loaded_model.predict(shapedStept);
            shapedStept = stepTPlus1
        all.append(shapedStept)
    return all
```

Figure 2: predictWithStateful(subset, startI, prevSteps, numIter, endI) is used to make time-series predictions.

After computing the time series predictions described above for 500 test scans, the correlation between the real scans and predicted scans was calculated. For instance, if numIter = 3, then we matched the real 103rd test scan with the value generated from our time series prediction using point 100. Since each sensor has a different reading and measures signals at different locations in the brain,

correlations were computed sensor-wise. So, once we got the 500 X 306 array of predicted values, we took the first column to get a 500 X 1 prediction vector for sensor 1. Similarly, the first column of the test subset was taken. We calculated the correlation between these two vectors by first using the scipy.stats.zscore() function to convert each vector to a vector of z scores, then used the numpy.corrcoef(z-score_real_array, z-score_predicted_array) to computed the Pearson correlation between these two vectors. A similar analysis was done for all 306 sensors.

Following this, for each sensor, we plotted the real sensor values and the predicted sensor values in order to gauge whether the predicted values were a reasonable fit to the actual values. Some of these graphs have been shown below, along with tables representing correlation values for a subset of the 306 sensors for number of time steps equal to 1, 2 and 3.

Note that in MEG scans, there are two different kinds of sensors: magnetometers and gradiometers. Gradiometers measure the amplitude of activation at a certain location in the brain, while MEGs measure the amplitude in the context of spatial orientation and axis. Typically, magnetometer data fluctuates widely and is much more noisy than gradiometer data.

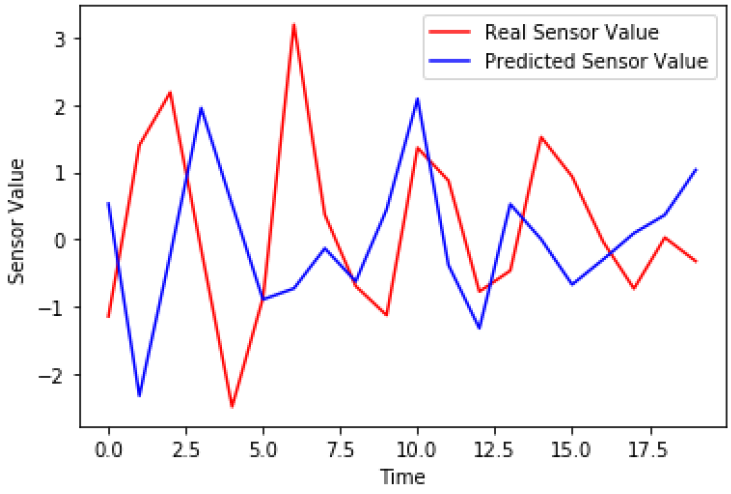

Figure 3: Real and predicted sensor values for sensor 1 (magenetometer sensor)

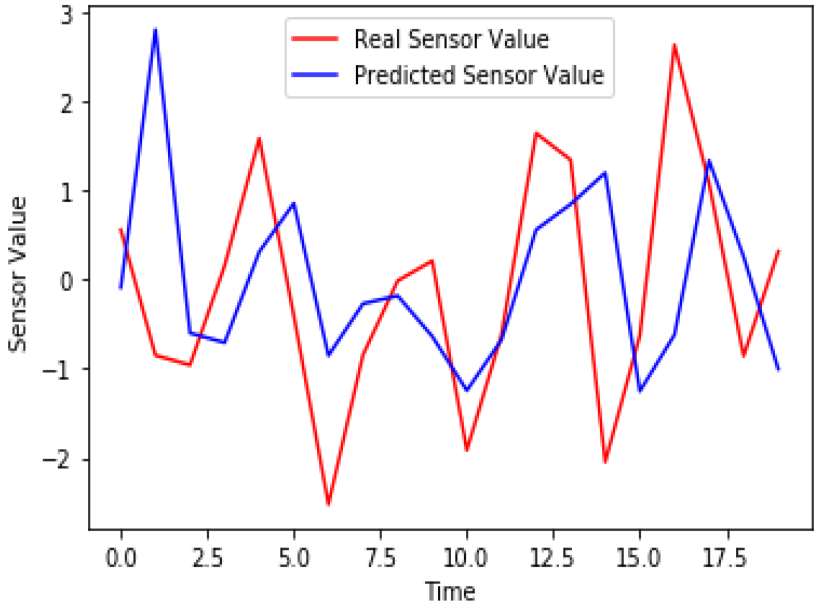

Figure 4: Real and predicted sensor values for sensor 3 (gradiometer sensor)

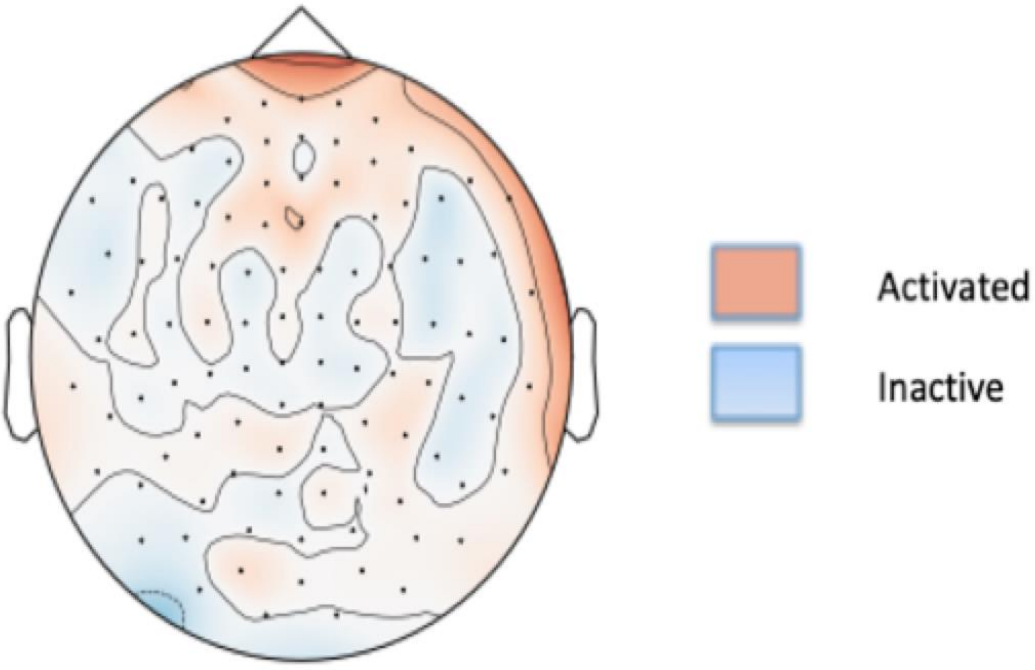

Figure 5: Sample MNE visualization for brain activation for brain scan number 2. Darker red areas correspond to areas of higher magnitude activation, as measured by the MEG.

	1 Timestep	2 Timesteps	3 Timesteps
Sensor 1	0.26	0.098	0.0034
Sensor 2	0.21	0.16	0.099
Sensor 3	0.17	-0.034	-0.15
Sensor 4	0.24	0.13	0.04
Sensor 5	0.05	-0.009	-0.05
Sensor 6	0.22	0.14	0.02

Table 1. Correlations between real and predicted values for some sensors. Note that correlations are moderate to high when predicting 1 time step in the future, but decrease for 2 and 3 time steps. Negative correlations are probably due to noise.

In the dataset, every third sensor starting from the first one is a magnetometer. So, sensors 1, 4, 7…. are magnetometers, and sensors 2, 3, 5, 6…. are gradiometers. In general, it was observed that the predicted curve approximated the real curve for sensor values more closely for gradiometer sensors than magnetometer ones. This corroborates the fact that magnetometer data is typically noisier and harder to predict than gradiometer data.

Lastly, we calculated correlations across corresponding time periods, for different words. For instance, if we consider the 500 X 1 prediction vector for each sensor as described above, we group these values by time. Since each word was presented to the subjects for 500 milliseconds each, and scans were taken 25 milliseconds apart, the values corresponding to the period 0-25 milliseconds for each word in this 500 X 1 vector are indices 0, 20, 40, and so on. Similarly,

values corresponding to 25-50 milliseconds are indices 1, 21, 41, etc. Pearson correlations were computed within each time group in order to determine whether prediction accuracy differs across time periods.

	0-25ms	25-50ms	50-100ms	100-125ms	125-150ms
Sensor 1	0.29	-0.34	-0.21	0.13	0.26
Sensor 2	0.40	0.06	0.28	-0.01	0.39
Sensor 3	0.51	0.12	-0.06	0.12	0.17
Sensor 4	-0.05	0.41	0.75	0.56	-0.33
Sensor 5	0.25	-0.01	0.14	0.01	0.54
Sensor 6	0.56	0.02	0.08	0.72	0.24

Table 2: Correlations between real and predicted sensor values, computed for words at the same time period of presentation to subject. More significantly high correlations (values above 0.4) observed. This is for 1 time step in the future.

In this setting, significantly higher correlations were observed. Additionally, correlations across words were observed to be higher in the first 250 milliseconds than the latter 250 milliseconds. This could be due to the way our brains react when exposed to words. In the initial period, there is a high burst of activity in visual cortex areas, corresponding to attempts to syntactically and semantically distinguish the word. This burst of activity is quite similar across words. Following this burst, after the word is interpreted, activations vary drastically across words based on sematic associations that trigger varied activations for different words.

Surprises and Lessons Leaned

Initially, when plotting graphs as shown in Figures 3 and 4, we did not distinguish between graphs for different kinds of sensors (magnetometer vs.

gradiometer). Thus, it was difficult to determine why certain sensors performed better than others. After assessing all the graphs, however, there appeared to be a clear trend that spurred me to look more closely at differences in the dataset. This made me more aware of intrinsic variability in datasets, and the importance of taking all possible factors affecting data readings into account before attempting to generalize results.

Results and Future Potential

It appears that brain activations can be predicted fairly well based solely on previous activations, using state-preserving LSTMs.

Due to noise, the inaccuracy of predictions is compounded when propagating for future time steps. Decreasing correlations as the number of prediction time steps in the future is increased evidences this. Using LSTMs for activation prediction thus seems to be mostly applicable only to the immediate next time step.

It appears that activations are also similar across words, for the same period of presentation to the subject, suggesting similarities in the way our brains process different words. This finding helps corroborate the theory of brain activations corresponding to different phases in textual exposure: the initial “understanding” phase and the subsequent “association” phase, and how the first phase is similar across words while the latter differs substantially.

Possible further areas of exploration include the analysis of these brain image predictions for semantic indicators. For instance, neural network models could potentially be trained to learn which brain scan sequence roughly corresponds to certain thoughts or words. Coupled with this study on the prediction of future brain activation states from previous activation states, this offers exciting opportunities to predict future thoughts of individuals based on

brain activity, providing widespread applicability to brain computer interface research, and potentially hands-free, voice-free communication and even the determination of intent based on brain activity.