EasyChair Preprint

№ 1774

# Mind-Reading AI : Re-Create Scenario from Brain Database

Poondru Prithvinath Reddy

EasyChair preprints are intended for rapid dissemination of research results and are integrated with the rest of EasyChair.

October 26, 2019

# Mind-Reading AI : Re-Create Scenario from Brain Database

**Poondru Prithvinath Reddy**

## ABSTRACT

An electroencephalography (EEG) based Brain Computer Interface (BCI) enables people to communicate with the outside world by interpreting the EEG signals of their brains. In their study, Xiang Zhang et all, proposed a novel deep neural network based learning framework that affords perceptive insights into the relationship between the EEG data and brain activities and designed a joint convolutional recurrent neural network that simultaneously learns robust high-level feature presentations through low-dimensional dense embeddings from raw EEG signals. The proposed approach has been to use results of this study as it is and use simulated conditions as true input for our study. We have developed a method called "deep text reconstruction," which uses a reconstruction algorithm capable of "decoding" a "hierarchy" of textual information from different sources, such as statements, facts, etc. Our algorithm also optimizes the output of the decoded text so that it more closely resembles the actual or true testimony, in combination with a multiple-layered feed forward neural network (NN) to simulate the same processes that occur when a human brain perceives language or text. The results show that our approach performs a baseline and the state-of-the art methods, yielding a good classification accuracy. The applicability of our proposed approach is further demonstrated with a practical system for crime detection.

## INTRODUCTION

Our minds may no longer be a safe haven for secrets. Scientists are working toward building mind-reading algorithms that could potentially decode our innermost thoughts through memories that act as a database.

A recent research study could give a voice to those who no longer have one. Scientists used electrodes and artificial intelligence to create a device that can translate brain signals into speech. Electrodes on the brain have been used to translate brainwaves into words spoken by a computer. This technology could help restore the ability to speak in people with brain injuries or those with neurological disorders such as epilepsy, Alzheimer disease, multiple sclerosis, Parkinson's disease and more

For the first time, this study demonstrated that we can generate entire spoken sentences based on an individual's brain activity, and we should be able to build a device that is clinically viable in patients with speech loss.

But these thought-decoding technologies are improving. A team of scientists based in Japan have now developed a new method called "deep image reconstruction," which uses a reconstruction algorithm capable of "decoding" a "hierarchy" of complex visual information from human brain activity, such as colors and shapes. The team's algorithm also optimizes the pixels of the decoded image so that it more closely resembles the actual object, in combination with a multiple-layered deep neural network (DNN) to simulate the same processes that occur when a human brain perceives an image.

A new study from scientists in Norway reveal how the brain keeps time and found the brain cells that place our memories in the right order. In the part of the brain called the lateral entorhinal cortex, they have found nerve cells that give each moment its distinctive signature and it is believed that this code keeps track of the order of events that happen. The code gives us a sense of time in relation to events. While further studies in understanding how the brain keeps track of time is progressing, scientists have now found traces of this understanding of temporality in the brains of the laboratory rats.

# METHODOLOGY

In this paper, we explain how our approach to this kind of AI-assisted translation of perceptual content better imitates the elaborate, hierarchical neural representations that are constructed in humans' natural code system.

We proposed a unified deep learning framework that leverages recurrent convolutional neural network to capture spatial dependencies of raw EEG signals based on features extracted by convolutional operations and temporal correlations through RNN architecture. Also, an Autoencoder layer is fused to cope with the possible incomplete and corrupted EEG signals to enhance the robustness of EEG classification. The results of their study, Xiang Zhang et all, has been used as it is and used simulated conditions as true input for our study.

We have developed a method called “deep text reconstruction,” which uses a reconstruction algorithm capable of “decoding” a “hierarchy” of textual information from different sources, such as statements, facts, etc. This data is then filtered through a deep neural network so that the decoding process would occur in a way that more closely matched what happens in the human brain when it perceives something. We considered Word2Vec with prediction based embedding technique CBOW( Continuous Bag of Words ) to generate and reconstruct word vectors. Our algorithm also optimizes the output of the decoded text so that it more closely resembles the actual or true testimony, in combination with a multiple-layered feed forward neural network (NN) to simulate the same processes that occur when a human brain perceives language or text.

## ARCHITECTURE

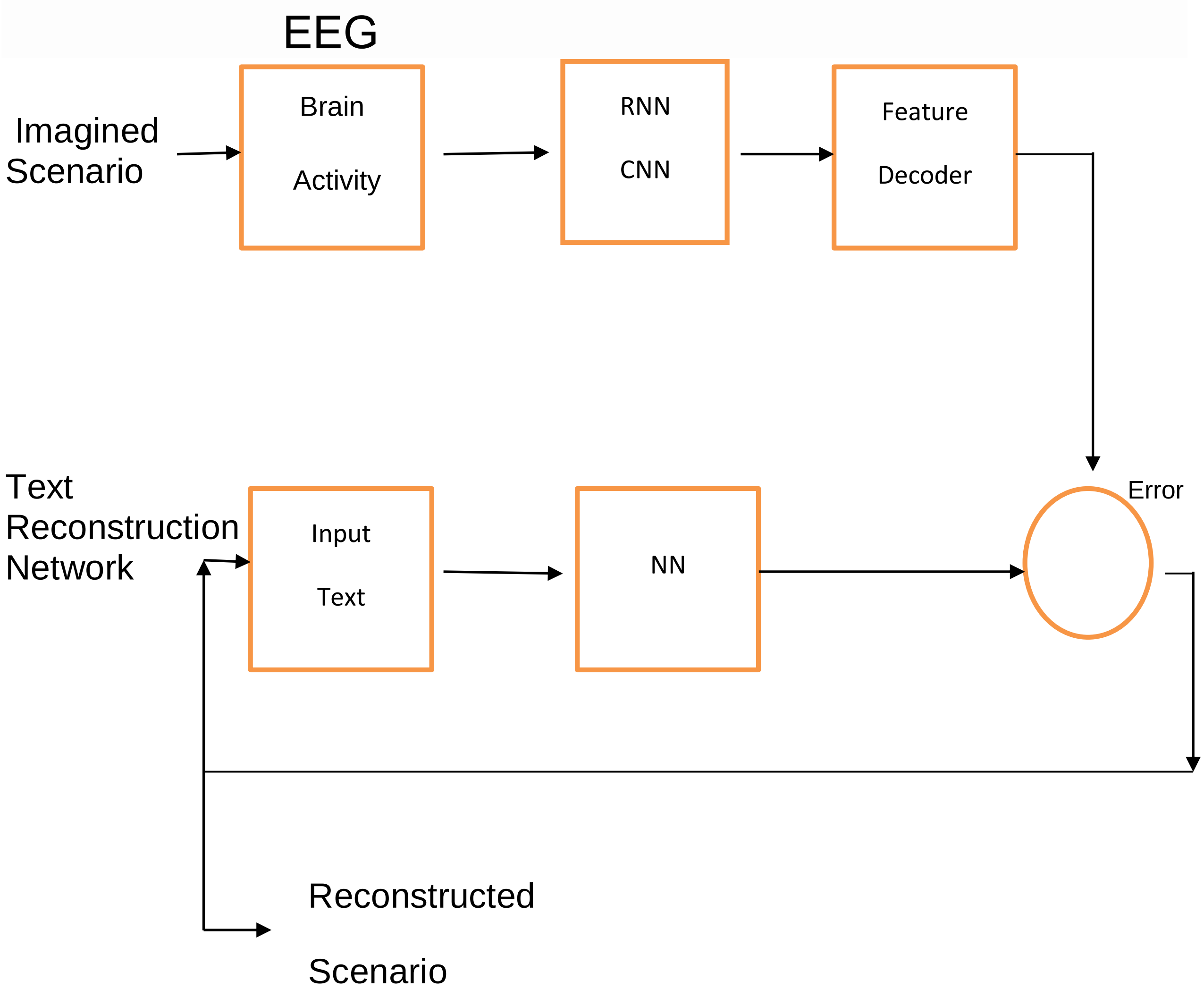



Figure 1 : Architecture for Reconstructing Scenario.

## CONVERTING EEG SIGNALS TO TEXT

An electroencephalography (EEG) based Brain Computer Interface (BCI) enables people to communicate with the outside world by interpreting the EEG signals of their brains to interact with the world. In a research paper, Xiang Zhang et all (https://arxiv.org/pdf/1709.08820.pdf ) proposed a novel deep neural network based learning framework that affords perceptive insights into the relationship between the MI-EEG( Motor Imagery EEG ) data and brain activities. They designed a joint convolutional recurrent neural network that simultaneously learns robust high-level feature presentations through low-dimensional dense embeddings from raw MI-EEG signals. They also employ an Autoencoder layer to eliminate various artifacts such as background activities. The proposed approach has been evaluated extensively on a large scale public MI-EEG dataset and a limited but easy-to-deploy dataset collected in their lab. The results show that the adopted approach outperforms a series of baselines and the competitive state-of-the art methods, yielding a classification accuracy of 95.53%. The applicability of their proposed approach is further demonstrated with a practical BCI system for typing.

The main offerings of this paper are highlighted as follows:

- Designed a unified deep learning framework that leverages recurrent convolutional neural network to capture spatial dependencies of raw EEG signals based on features extracted by convolutional operations and temporal correlations through RNN architecture, respectively. Moreover, an Autoencoder layer is fused to cope with the possible incomplete and corrupted EEG signals to enhance the robustness of EEG classification.

- Evaluated extensively the model using a public dataset and also a limited but easy-to-deploy dataset that was collected using an off-the-shelf EEG device. The experiment results illustrate that the proposed model achieves high level of accuracy over both the public dataset (95.53%) and the local dataset (94.27%). This demonstrates the consistent applicability of the proposed model.

- Also presented an operational prototype of a brain typing system based on the proposed model, which demonstrates the efficacy and practicality of adopted approach.

The proposed model consists of a design - an RNN model consisting of three components: one input layer, 5 hidden layers, and one output layer. There are two layers of Long Short-Term Memory (LSTM) cells among the hidden layers. While RNN is good in exploring the temporal (inter-sample) relevance, it is unable to appropriately decode spatial feature (intra-sample) representations. To exploit the spatial connections between different features in each specific EEG signal,   a CNN structure is designed. The CNN structure is comprised of three categories of components: the convolutional layer, the pooling layer, and the fully connected layer. The convolutional layer contains a set of filters to convolve with the EEG data and then through the feature pooling and non-linear transformation to extract the geographical features. CNN is well-suited to extract the spatial relevance of the 2-D input data efficiently.

Next, a feature adaptation method is designed to map the stacked features to a correlative new feature space which can fuse the temporal and spatial features together and highlight the useful information. To do so, an Autoencoder layer is introduced to further interpret EEG signals, which is an unsupervised approach to learning effective features. The Autoencoder is trained to learn a compressed and distributed representations for the stacked EEG feature X'. The input of Autoencoder is the stacked temporal and spatial feature X'. Assume h, $\mathring{X}'$ denote the hidden layer and output layer data, respectively.

The data transformation procedure is described as the following:

$$h = Wen\ X' + ben$$

$$\mathring{X}' = Wde\ h + bde$$

where Wen, Wde, ben, bde denote the weights and biases in the encoder and decoder.

# CONVERTING TEXT TO SCENARIOS

As it turns out, many Machine Learning algorithms and almost all Deep Learning Architectures are incapable of processing *strings* or *plain text* in their raw form. They require numbers as inputs to perform any sort of job, be it classification, regression etc.  And with the huge amount of data that is present in the text format, it is imperative to extract knowledge out of it and build applications. Some real world applications of text applications are – sentiment analysis, document or news classification, etc.

Let us now define Word Embeddings formally. A Word Embedding format generally tries to map a word using a dictionary to a vector. There are different types of Word Embeddings or Word Vectors and We have considered  Prediction based Embedding because of its advantages as the method  provides probabilities to the words and useful for tasks like word analogies and word similarities.

So let us look at the word2vec model used as of today to generate word vectors.

Word2Vec is a shallow, two-layer neural networks which is trained to reconstruct linguistic contexts of words. Word2Vec is a simple neural network with a single hidden layer, and like all neural networks, it has weights, and during training, its goal is to adjust those weights to reduce a loss function.

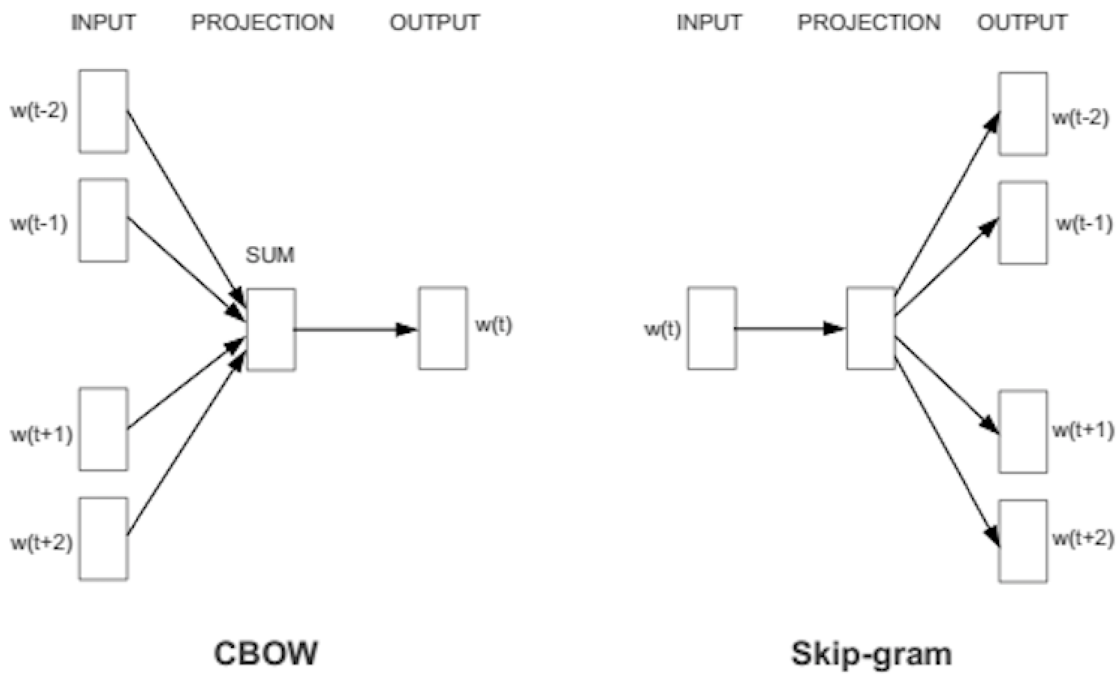
It takes as its input a large corpus of words and produces a vector space, typically of several hundred dimensions, with each unique word in the corpus being assigned a corresponding vector in the space.
Word vectors are positioned in the vector space such that words that share common contexts in the corpus are located in close proximity to one another in the space. Word2Vec is a particularly computationally-efficient predictive model for learning word embeddings from raw text.

The architecture is similar to an autoencoder’s one, you take a large input vector, compress it down to a smaller dense vector and then instead of decompressing it back to the original input vector as you do with autoencoders, you output probabilities of target words.

First of all, we cannot feed a word as string into a neural network. Instead, we feed words as one-hot vectors, which is basically a vector of the same length as the vocabulary, filled with zeros except at the index that represents the word we want to represent, which is assigned “1”.

The hidden layer is a standard fully-connected (Dense) layer whose weights are the word embeddings. The output layer outputs probabilities for the target words from the vocabulary. The rows of the hidden layer weight matrix, are actually the word vectors (word embeddings) we want. The hidden layer operates as a lookup table. The output of the hidden layer is just the “word vector” for the input word. The end goal of all of this is to learn this hidden layer weight matrix and then toss the output layer when we’re done. The output layer is simply a softmax activation function:

Word2Vec comes in two flavors, and we have used the Continuous Bag-of-Words (CBOW) model in this study.
Algorithmically, the model is as below:

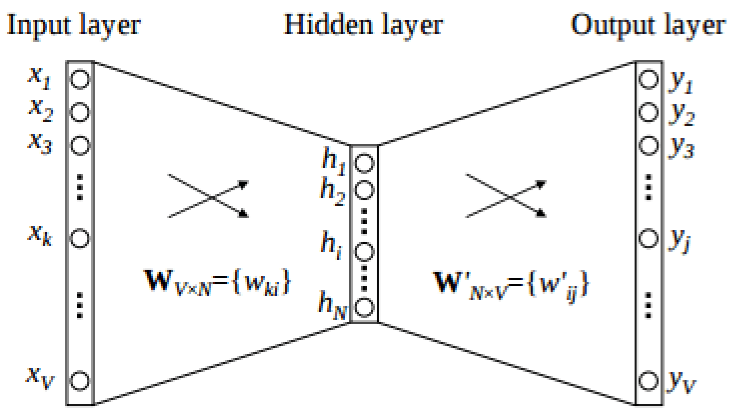



## CBOW (Continuous Bag of words)

The way CBOW work is that it tends to predict the probability of a word given a context. A context may be a single word or a group of words.

The input vector is sent into a shallow neural network with three layers: an input layer, a hidden layer and an output layer. The output layer is a softmax layer which is used to sum the probabilities obtained in the output layer to 1. Now let us see how the forward propagation will work to calculate the hidden layer activation.

Let us first see a diagrammatic representation of the CBOW model.




The matrix flow of the above image is as below.

1. The input layer and the target, both are one- hot encoded.
2. There are two sets of weights. One is between the input and the hidden layer and second between hidden and output layer.
3. There is a no activation function between any layers.( More specifically, referring to linear activation)
4. The input is multiplied by the input-hidden weights and called hidden activation. It is simply the corresponding row in the input-hidden matrix copied.
5. The hidden input gets multiplied by hidden- output weights and output is calculated.
6. Error between output and target is calculated and propagated back to re-adjust the weights.
7. The weight  between the hidden layer and the output layer is taken as the word vector representation.

In case of multiple context word, only the calculation of hidden activation changes. Instead of just copying the corresponding rows of the input-hidden weight matrix to the hidden layer, an average is taken over all the corresponding rows of the matrix. The average vector calculated becomes the hidden activation. So, if we have three context words for a single target word, we will have three initial hidden activations which are then averaged element-wise to obtain the final activation.

## Using pre-trained word vectors

We have used google's pre-trained model for word vectors. It contains word vectors for a vocabulary of 3 million words trained on around 100 billion words from the google news dataset.

# EXPERIMENT

The objective of this experiment is to detect crime in investigations and to predict the crime on the test dataset.

In the first part , a police investigator peeks inside the head of potential witnesses using a EEG AI-powered device. The small device enables the investigator to see through the memories of the witnesses by projecting text to a separate display machine. This text about crime which is displayed on a machine by reading brain of witnesses is taken as true evidence or proof in accordance with fact.
In this paper, we have not implemented converting EED signals to Text but taken the results of the study by Xiang Zhang et all (
https://arxiv.org/pdf/1709.08820.pdf
) as the basis. However, the results of the study are simulated into different crime scenarios to test our architecture of mind-reading AI.

In the second part, we have built around facts, evidence, interrogation and statements of many people including the victims which are fed as input text into NN to arrive at crime summary or to determine the whole truth from a given testimony. And this summarized output is compared with testimony given by eye-witnesses( EEG signals ) which is considered as true evidence and any error is fed back into the NN as input text to improve the accuracy. Also we have tested our crime reconstruction network with different and not the same events around facts, evidence, interrogation, statements,etc.

# RESULTS

The results of the study are not encouraging with low accuracy of summarized crime information in comparison to evidence given by eye-witnesses regarding crime scenario. This is due to the fact that the training of Neural Network for reconstruction was done with vectorised information from Google rather than the actual or real crime dataset.

# CONCLUSION

In this paper, we present a hybrid deep learning model to decode the raw EEG signals for the aim of converting the user's thoughts to texts. The model employs the RNN and CNN to learn the temporal and spatial dependency features from the input EEG raw data and then stack them together. Our proposed approach adopts a method called "deep text reconstruction," which uses a reconstruction algorithm capable of "decoding" a "hierarchy" of textual information from different sources, such as statements, facts, etc. Our algorithm also optimizes the output of the decoded text so that it more closely resembles the actual or true testimony. We evaluate our approach on a Google dataset and not on a real world dataset . The results are encouraging and form the basis for state-of-the-art method for crime detection.

# REFERENCE

Xiang Zhang, Lina Yao, Quan Z. Sheng, Salil S. Kanhere, Tao Gu, Dalin Zhang "Converting Your Thoughts to Texts: Enabling Brain Typing via Deep Feature Learning of EEG Signals" , https://arxiv.org/pdf/1709.08820.pdf