Intel-PNI Collaboration

Decoding of generic mental representations from fMRI November 6, 2015

Lecturer: Francisco Pereira Scribe: Kiran Vodrahalli

Contents

1 Overview

This work extends Pereira et. al's approach in their 2011 paper in the following ways:

1. Replace generative model with an RNN to generate grammatically sensible outputs. Alternatively, you can use something other than an RNN since there are two separate generation tasks (the gist of the passage and the natural language representation of each sentence).

2. Word vectors are not represented by topic probability distributions, but rather by a word-vector of your choice (analogous to word2vec or GloVe). In particular, they use GloVe and are looking into using word vectors which capture a single sense of the word.

3. A key difference is that they create a different dataset: They use an approach that partitions the vector space into 180 regions containing vectors for words that are used similarly in corpora (e.g. “happy” and “sad” would both be used in discussions of emotion). We then choose $1+5$ words from each region (1 is a good representative or a word naming what the words have in common, 5 are other good ones). Looking at the the vectors for representative words *a posteriori*, it turns out that each dimension of the semantic space is used by $10-20$ words over the 180; while this is not guaranteed, it was something they hoped this approach would lead to (and it does, for various different word representations).

4. On the training tasks, you have either 6 sentences using the representative word and some others, vector, 6 pictures paired with each representative word, or a wordcloud with the representative at the center, surrounded by five others. These representations are associated with each of the 180 concept/word-vectors. Then, using representations of these various things (sentences, word-pictures, wordcloud) in the form of the concept/word vectors corresponding to the stimulus (sentences, word-pictures, wordcloud) as input, you predict a voxel activation

for each voxel in the brain with ridge regression. For each task, you go backwards (by effectively doing convex optimization to optimize the word vector for a given brain activation, given that you have learned some matrix C where each column is voxel-vector associated with a given word-vector dimension? i.e., $Y = CX$, where Y = fMRI, C = learned matrix, X = word vectors you train with ridge regression given (X, Y). Then after learning C, given a novel Y, you optimize to get X). The optimization is slightly more involved (namely, an SVD is thrown into the mix to do dimension reduction).

2 Introduction

This work has been partitioned into doing imaging at Princeton and MIT, and Siemens for various aspects of the data processing and so on. Funding is from IARPA and the Knowledge Representation group.

3 Generative Brain Decoding

If you want to actually try to understand how the brain is understanding something, you want to learn what the brain would build while a person carries out a task. This is from Naselaris 2009 paper. This forward model can be validated in two ways: by predicting fMRI activation for a novel stimulus, or by predicting the stimulus. There are a lot of examples of similarity.

IARPA would like any input (text, pictures, sound, $\cdots$), any mental content, multiple time scales, and something that generalizes across sessions. So our practical research questions are: how to represent generic mental content, and what to generate as decoding output.

IARPA gives a test passage: three to five sentences, collect imaging data from that, get a gist of the meaning, a human interpretable representation of the gist (use Mech Turk), then try to pull out the exact sentences shown to the subject. This is somewhat business, and they don't expect us to succeed wildly, but I hope to convince you that we are well on our way to doing so.

Our approach is to show the subjects stimuli that cover as much as we can of the semantic space so that our system can actually genearlize to any topic we have seen before up to a certain point. Then this has both words and sentences represented as semantic vectors, and a mapping of semantic dimensions to patterns of brain activation (basis images). Then we put this through a vector decoder through a language decoder (a recurrent neural net) or a concept decoder (a probability distribution).

4 Building the forward model and decoding

A word is represented as a vector in a semantic space. We have a ton of work comparing off the shelf representations. Global vectors are the one that do best, comparison paper comoing soon. For sentences, we represent them in the same semantic space of the words, which may or may not be a good assumption. The results I will show you were obtained by averaging the content words in a sentence - it is not the best, but it is embarrasingly good, so it is reasonable to use it here. We also have looked at paragraph vector and skip-thought (RNN method). We think skip-thought may be good (from Toronto).

If we have this mapping, we can look for what is common between the semantic vectors for different words. If words share a particular semantic dimension, we can ask what parts of the image are shared for each semantic dimension. Imagine that one dimension captures the fact that something can be manipulated like a tool, you would expect it to relate to motor cortex. Then you can cast things in terms of matrix factorization: $X = ZB$ where X is a brain activation and Z are semantic vectors and B is the basis. You do voxel-wise ridge regression to learn basis vectors. Given that you have more columns that examples, you need regularization. You can also just take SVD and work in a low-dimensional representation and modify the hardware to take that into account. You can do this in situations when you have many more dimensions than you have examples to work with. We have a way of going back and forth between brain images and semantic vectors. Is there a way to validate assumption about linearity? It is the easiest thing to use. There are nonlinearities we could exploit (check paper in NIPS from 2012 on local filters). Are dimensions of word vectors supposed to be interpretable? Generally, people do not try to do interpretation. But there is some work regarding sparsity, etc.

5 Imaging data

We need to collect data for multiple purposes: Learn a mapping between semantic space and fMRI, scan subjects during reading text passages for evaluation and testing. For mapping, we would use word stimuli in a naturalistic context, not in isolation. We want to maximize repetitions. We can show the word highlighted in different sentences, show the word we are interested in with a cloud of words, or show a word together with different relative images (closer to Mitchell in 2008). There is a benchmark paper from 2015 from Marco Beroni; these semantic vectors are related to relatedness, not similarity (analogies, similarity, etc). We are also publishing our own comparison at the end of the year.

We also need to cover the semantic space. We also want to keep learning to a single scanning session. With those constraints: We start with a vocabulary of $30,000$ words from a paper by Mark Brisbayer. We take these words and run spectral clustering which give us 200 clusters: words in terms of their vectors, or words similar to how they are used in a text corpus. These words tend to have a theme. For instance, taste and smell, various kinds of food, etc. In addition to clusters that correspond to classic categories. We also have states of mind and emotions. It does not mean that the words are similar, it means they occur in similar contexts. As you increase the number of clusters, you see unusual things: verbs that qualify how subject moves after it is hit, etc. In one session we take 180 stimulus, and select a representative word from each cluster. Here we avoid showing words with their multiple meanings (each word which is polysemous only is given in one sense). The words we pick use all of the dimensions of the semantic space - this is critical and allows you to generalize. You should have each dimension used by $10 - 20$ words. That way when you learn basis images you learn enough to estimate them. In the end, you have 6 sentences for each of the 180 concepts we picked and 6 pictures per thing. We have passages from (2011work) done of 24 categories, which we treat as topics.

6 Results

We look at which voxels replicate across sessions. What voxels represent things similarly across voxels. Let's look at the correlations between voxels in pictures and text. The majority of voxels do not replicate whatsoever between one sentence in one session and a sentence in another one. There is no decoding yet. You are just looking at the β from GLM (regression) on your data. There is some reliability, but less than expected. Across sentences $(0.8 - 0.85)$ accuracy using weights from first session on another session (identical); 0.9 for pictures.

For voxel consistency, we found that the voxels which respond more to semantic tasks. What we ended up finding was a way of selecting voxels which use those specific voxels to filter the brain to a smaller number of voxels. Out of those, we select voxels which are task-reponsive. We use those voxels to learn a basis and do de-coding. If we are doing classification, we get 87% accuracy (which passage are you reading); 80% for rank accuracy on 384 sentences. If we sort vectors by category, we see that the vectors correspond well to the actual text. This gives us hope we can generate text.

We can pull out semantic vectors when people pull out passages or sentences. Now we want to generate: Well, we can use RNN for caption generation. So our idea is to use the same approach to generate text: We take a sentence, represent it as a semantic vector, then learn to tag skip-thought vectors. Then with a good decoder, the idea is to plug in our vectors and decode something good.

These language generation models can predict directly from a semantic vector. This is basically an autoencoder. On pure text it works decently well on image captions. We are trying to get it to work on Wikipedia. Nobody has trained these models on this complicated Wikipedia vocabulary. We are also trying to find nearest words in semantic space (sentence-wise).

We can also generate a probability distribution over concepts. We need a universe of possible concepts to consider. Given we know what passages they are considering, we could have topics themselves involved, concepts relevant to understanding, related, or common understanding concepts. We use Wikipedia as an ontology. We can turn all of these into vectors. We are basically learning topic models of the Wikipedia corpus and use them to directly produce probability distributions over articles.

7 Conclusion

We want to build something that can satisfy any constraints. Generating language output directly are reasonable outputs to have for this kind of decoder. Learning a basis in this way is reasonable.