Course reader: *Spatiotemporal structure using forward models*

- The word model has several meanings: a person who wears clothes or jewelry, or a toy that is shaped like a large object (like a model airplane).

- In engineering and science, a **model** is a set of mathematical equations that represents a system in a way that captures the key parts of the system. Models are used all the time, because the real world is too complex, so simplified versions of it are used for research, development, testing, and so on.

- A **forward model** is a mapping from some unobservable hidden state to a set of measurements that can be simulated. This mapping is based on assumptions about the system that is being modeled. The idea is that complex systems are governed by complex internal dynamics that you cannot measure. So you use the data that you can measure and then build a model that (1) captures the important features of the measurements using (2) internal dynamics that you can explicitly manipulate in a computer, the way you couldn't manipulate them in teh real world.

- Typically the internal dynamics are low dimensional while the data that they produce are high-dimensional.

- In the video I gave two examples of climate change and weather dynamics. Another example could be models of the financial system. Financial analysts develop models of stock markets and then can control things like "uncertainty" and "speculation" that cannot be explicitly manipulated in the real world (obviously, those models leave some room for improvement, otherwise we wouldn't have stock market crashes every decade...).

- In my field of neuroscience, forward models are used to test new data analysis methods. The brain is unfathomably complex, and brain data are also really complex. We build simplified models of brain electrical activity that contain enough biophysical realism to validate analysis methods while being simple enough to control easily with a few parameters (don't worry, these models do not think or behave, and they are not going to take your job or tell you how to invest your money!).

- It's useful to start with simpler forward models that mix images and time series data. But don't be fooled by the simplicity of the geometric shape forward model examples: Even these models can lead to difficult problems in dimensionality reduction and source separation.

- I use some advanced techniques in this section, like singular value decomposition (SVD) and generalized eigendecomposition (GED), without explaining how or why they work. Don't be concerned if you are unfamiliar with these concepts. They are advanced linear algebra techniques for dimensionality reduction (also called compression) and source separation. I think it's nice to end a course with a taste of what you would learn if you continue in your education about data science, statistics, and machine learning. I hope you find these examples inspiring!

Exercises

1. One method of denoising a noisy dataset is by averaging over time. This methods works well if the noise is roughly normally distributed with mean=0. Implement this in the first example in the code by making a movie (after the `allimgs` matrix is already created) in which each frame is the average of 2 frames previous to 2 frames afterwards. This is called mean-smoothing with the parameter k=5. How does this movie compare to the original noisy version?

2. The parameter in this forward model is the brightness (amplitude) of the Gabor patch. Adjust the code to manipulate the width while leaving amplitude fixed. Once you get this idea, then you are capable of manipulating any other feature or combination of features.

3. Apply the SVD code to the data created in the first video. How many components contain signal?

4. In the "one active dipole" code, pick different dipoles to see what the resulting topographical map looks like. There are 2,004 dipoles, so you can pick any number between 1 and 2004. You'll probably want to pick different scalp channels to plot the time course from the channel close to the forward model projection.

5. The "two active dipoles" code uses a sine wave and a Gaussian as the two time series. Use some different kinds of time courses that you learned about in the section about simulating time series. Does the source separate code work equally well? Does that depend on the amplitude of the signal? (Add your new signals only after time point 500 like in the code.)

Answers

1. It looks pretty good. Definitely less noise, and it's easier to see the signal. But the noise is still there. Here's code to implement it.

PYTHON

```
k = 2
for i in range(k,n-k):
tmp = np.mean(allimgs[i-k:i+k,:,:],axis=0)
pl.cla() # wipe the figure
plt.imshow(tmp,vmin=-1,vmax=1)
display.clear_output(wait=True)
display.display(pl.gcf())
time.sleep(.0001)
```

MATLAB

```
k = 2;
for i=k+1:n-k
   tmp = squeeze( mean(allimgs(i-k:i+k,:,:),1) );
   set(h,'CData',tmp)
   pause(.05)
end
```

2. Add the following two lines of code inside the loop just above the `allimgs=...` line. The result looks pretty neat! (Note that this code does not include noise; you can add that if you want.)

PYTHON

```
formod = np.multiply(sine2d,gaus2d>ampmod[i])
imgtmp = formod/np.max(formod)
```

MATLAB

```
formod = sine2d .* (gaus2d>ampmod(i));
imgtmp = formod./max(formod(:));
```

3. All of the meaningful dynamics are contained in the first component. You have to modify the code in the first cell to get `allimgs` to be a time×pixels matrix similar to what's done in the second cell, then you can just re-run the SVD cell. You'll see that the first component captures all the dynamics (including the amplitude modulator) while the second one is just noise.

4. There isn't really a question here to answer. It's just interesting to play around with the code to get a better sense of how the forward model behaves.

5. The source separation should work for most time series, as long as they are different from each other and have sufficient amplitude. If the amplitude is too small, then the signal is indistinguishable from the noise.