Information Theory & Neural Coding

Total points 25

1. Suppose that we have a neuron which, in a given time period, will fire with probability 0.1, yielding a *Bernoulli distribution* for the neuron's firing (denoted by the random variable F = 0 or 1) with P(F = 1) = 0.1.

2 points

Which of these is closest to the entropy H(F) of this distribution (calculated in bits, i.e., using the base 2 logarithm)?

- 0.4690
- -0.1954
- 1.999
- 0.1954

2. Continued from Question 1:

4 points

Now lets add a stimulus to the picture. Suppose that we think this neuron's activity is related to a light flashing in the

If there is a flash, the neuron will fire with probability $1/2$. If there is not a flash, the neuron will fire with probability $1/18$. Call this random variable $\vec{F}$ (whether the neuron fires or not).

Which of these is closest, in bits (log base 2 units), to the mutual information MI($\vec{S}$,$\vec{F}$)?

- 0.0904
- 0.8476
- 0.3786
- -0.3786

3.

1 point

$$I(\vec{x}) = \sum_i a_i \, \phi_i(\vec{x}) + \epsilon(\vec{x})$$

$$E = \sum_{\vec{x}} \left[I(\vec{x}) - \sum_i a_i \, \phi_i(\vec{x}) \right]^2 + \lambda \sum_i C(a_i)$$

This math from lecture 4.3 could potentially be intimidating, but in fact the concept is really simple. Getting an intuition for it will help with many types of problems. Let's work out a metaphor to understand it.

Suppose we want to build a complex image. We could do that by layering a whole bunch of pieces together (mathematically - summing). This is like drawing on transparencies with various levels of opacity and putting them

Of course, if our neural system was trying to make visual fields that worked for any sort of input, we would want more than just roads, skies, and bikes to work with! One possibility is to have a bunch of generic shapes of various sizes, orientations, and locations within the image. If we chose the right variety, we could blend/sum these primitive pieces together to make just about any image! One way to blend them is to let them have varying transparencies/opacities, and to set them on top of each other. That is what we would call a weighted sum, where the weights are how transparent each piece is.

Of course, we may not want to have too many possible shapes to use. As mentioned in the video, the organism likely wants to conserve energy. That means having as few neurons firing as possible at once. If we conceptually make a correlation between these shapes and the neurons, then we can point out we would want to use as few shapes as we could while maintaining an accurate image.

This math gives us a way of summing a bunch of pieces together to represent an image, to attempt to make that representation look as much like the image as possible, and to make that representation efficient - using as few pieces as possible. That is a lot of work for two lines of math!

Now let's put this metaphor into action to understand what all these symbols mean. I'll give you one to start with. The vector x in the equation above represents the coordinates of a point in the image. Now you fill in the rest:

What do the ϕ_i's, called the "basis functions," represent in our metaphor?

- The level of transparency vs. opacity/influence of each piece.
- The importance of coding efficiency.
- The difference between the actual image and the representation.
- The pieces that make up the image.

What does ϵ represent?

- The difference between the actual image and the representation.
- The importance of coding efficiency.
- The level of transparency vs. opacity/influence of each piece.
- The pieces that make up the image.

5. Continued from Question 7:

1 point

What do the a_i's represent?

- The level of transparency vs. opacity/influence of each piece
- The importance of coding efficiency.
- The pieces that make up the image.
- The difference between the actual image and the representation.

6. Continued from Question 3:

1 point

What does λ represent?

- The pieces that make up the image.
- The importance of coding efficiency.

7. In the following three questions, we will explore Poisson neuron models and population coding.

5 points

This exercise is based on a set of artificial "experiments" that we've run on four simulated neurons that emulate the behavior found in the cercal organs of a cricket. Please note that all the supplied data is synthetic. Any resemblance to a real cricket is purely coincidental.

In the first set of experiments, we probed each neuron with a range of air velocity stimuli of uniform intensity and differing direction. We recorded the firing rate of each of the neurons in response to each of the stimulus values. Each of these recordings lasted 10 seconds and we repeated this process 100 times for each neuron-stimulus combination.

We've supplied you with a .mat file for each of the neurons that contains the recorded firing rates (in Hz). These are named `neuron1`, `neuron2`, `neuron3`, and `neuron4`. The stimulus, that is, the direction of the air velocity, is in the vector named `stim`.

Download the file:

tuning
MAT File

and save it into your MATLAB/Octave directory. To load the data, use the following command:

tuning_2.7
PICKLE File

tuning_3.4
PICKLE File

To load the data, make sure you are in the same directory you saved it and add the following (shown for 2.7) to your script:

```
1  import pickle
2
3  with open('tuning_2.7.pickle', 'rb') as f:
4      data = pickle.load(f)
5
```

This will load everything into a dict called data, and you'll be able to access the stim and neuron responses using data['stim'], data['neuron1'], etc. (In general, data.keys() will show you all the keys available in the dict.)

(NOTE ON DOWNLOADING CODE AND DATA: Currently, downloaded files are automatically renamed to begin with a long string of random characters (we hope to have this fixed soon). Sometimes the file type is also changed. In order to ensure that all of the files in the quizzes work correctly, make sure that after downloading each file you rename it to the file name shown in the original quiz question. If you still have trouble getting any of the files to open feel free to search or inquire on the class Discussion Forums.)

the stimulus of value stim(n) to `neuron1`.

Plot the tuning curve-- the mean firing rate of the neuron as a function of the stimulus--for each of the neurons.

Which of the following functions best describes the tuning curve?

- Linear function.
- Unrectified cosine.
- Gaussian.
- Half-wave rectified cosine.

8. Continued from Question 7:

5 points

We have reason to suspect that one of the neurons is not like the others. Three of the neurons are Poisson neurons (they are accurately modeling using a Poisson process), but we believe that the remaining one might not be.

Which of the neurons (if any) is NOT Poisson?

Hint: Think carefully about what it means for a neuron to be Poisson. You may find it useful to review the last lecture of week 2. Note that we give you the *firing rate* of each of the neurons, not the spike count. You may find it useful to convert the firing rates to spike counts in order to test for "Poisson-ness", however this is not necessary.

In order to realize why this might be helpful, consider the fact that, for a constant a and a random variable X, $\mathbb{E}[aX] = a\mathbb{E}[X]$ but $Var(aX) = a^2 Var(X)$. What might this imply about the Poisson statistics (like the Fano factor) when we convert the spike counts (the raw output of the Poisson spike generator) into a firing rate (what

- Neuron 3.
- None.
- Neuron 2.
- Neuron 1.

9. Continued from Question 7:

5 points

Finally, we ran an additional set of experiments in which we exposed each of the neurons to a single stimulus of unknown direction for 10 trials of 10 seconds each. We have placed the results of this experiment in the following file:

pop_coding
MAT File

You should save the file into your MATLAB/Octave directory and import the data using the following command:

load('pop_coding.mat')

The equivalent python files are:

pop_coding_2.7
PICKLE File

These can be loaded in the same way as described in question 7 above.

(NOTE ON DOWNLOADING CODE AND DATA: Currently, downloaded files are automatically renamed to begin with a long string of random characters (we hope to have this fixed soon). Sometimes the file type is also changed. In order to ensure that all of the files in the quizzes work correctly, make sure that after downloading each file you rename it to the file name shown in the original quiz question. If you still have trouble getting any of the files to open feel free to search or inquire on the class Discussion Forums.)

corresponding to neuron 1, neuron 2, neuron 3, and neuron 4.

Decode the neural responses and recover the mystery stimulus vector by computing the population vector for these neurons. You should use the maximum average firing rate (over any of the stimulus values in 'tuning.mat') for a neuron as the value of r_{max} for that neuron. That is, r_{max} should be the maximum value in the tuning curve for that neuron.

What is the direction, in degrees, of the population vector? You should round your answer to the nearest degree. Your answer should contain the value only (no units!) and should be between 0° and 360°. If your calculations give a negative number or a number greater than or equal to 360, convert it to a number in the proper range (you may use the mod function to do this).

You may need to convert your resulting vector from Cartesian coordinates to polar coordinates to find the angle. You may use the atan() function in MATLAB to do this. Note that the the convention we're using defines 0° to point in the direction of the positive x-axis and 90° to point in the direction of the positive x-axis (i.e., 0 degrees is north, 90 degrees is east).

Enter answer here

Coursera Honor Code Learn more

I understand that submitting work that isn't my own may result in permanent failure of this course or deactivation of my Coursera account.

Daniel Engbert

Use the name on your government issued ID

Submit | Save draft