Chapter 9

Analysis of population data

Given the vast numbers of neurons involved in cognition and information processing, it is perhaps surprising that observation of a single neuron's activity has ever yielded much insight into brain function. The focus through much of the history of systems neuroscience has been on the analysis of those neurons whose firing rates change strongly and reliably in a particular direction, either in response to a stimulus, or as an indication of a forthcoming motor response. This focus arose in part because of limitations of the experimental apparatus available—when recordings of individual neurons are hard to come by, it is important to find and select those neurons with the strongest task-dependent response. Moreover, the observation of a distinct change in a neuron's firing rate is both easier to explain and more dramatic—therefore, more newsworthy and publishable—than any correlation with behavior extracted more painstakingly from changes that are subtle if observed at the level of the individual neuron. Therefore, the desire to find individual neurons with strong, task-specific responses continues today.

Models of neural circuits—such as the majority of those presented in this book—are similarly focused on explaining the strong, task-dependent responses of certain neurons. In part, such models are easier to formulate and understand than those in which the encoding of information is almost impossible to pinpoint. In this book, the one network we have considered with near impenetrable encoding is the network of granule cells in the cerebellum, which in our model (Tutorial 8.4) encoded time. While the network activity can produce a temporally precise change in Purkinje cell activity, we would be hard-pressed to find a "time-tuned" neuron in the network. It is quite likely that much neural activity beyond the primary sensory and primary motor areas of any animal's brain is of this ilk[1-9]. We spend less time on such circuits in this introductory book, simply because they are far from being understood and they are more difficult to simulate (see [5,10-16] for examples and further reading)—not because they are less important for brain function.

The first step toward understanding the mechanisms, or even the general principles and algorithms involved, when they are not revealed by straightforward changes in the firing rates of individual neurons, is to employ appropriate methods to combine activity from many cells[17-21]. In this chapter, we consider three such methods of analysis, each of which can be considered the simplest prototype of a large set of related methods.

9.1. Principal component analysis (PCA)

Box 9.1. Principal component analysis (PCA) is a method for reducing high-dimensional data to fewer dimensions in a manner that aims to minimize loss of information.

Box 9.2. Dimensionality of data is the number of distinct values, or coordinates, used to define one data-point. The dimensionality can be the number of variables in the system, or the number of variables multiplied by the number of time-points measured.

Before describing PCA, it is important to understand the general concept of "high-dimensional data" and why "dimensionality reduction" can be useful. The ideas are of general applicability and value to multiple fields of data analysis, ranging from the sorting of spikes (described in the next subsection), to the unbiased scoring of elected officials by analysis of their voting records.

The high dimensionality refers to the number of values used to characterize a data-point. For example, a particular curve, such as a voltage trace, $V(t)$, can be characterized by a large number of sampled points along the curve. In this manner, a curve characterized by 100 points along the trajectory can be represented by a single point in 100-dimensional space. A dataset with many such curves would then be represented by one or more clusters of such points in the 100-dimensional space.

Similarly, the complete set of votes on a resolution can be characterized by a set of values, one value for each legislator voting. For example, votes in favor of, or against the resolution, or abstentions, can be ascribed values of "+1" or "-1" or "0" respectively. In this manner, the votes of 100 senators would be represented by a single point in 100-dimensional space. A dataset containing many such voting outcomes, would then be represented by one or more clusters of such points in the 100-dimensional space, just as in the previous example.

PCA is one of several methods for extracting useful information from such clusters of data points in high-dimensional space. At the heart of PCA is the ideas of choosing a basis in geometry—which means choosing a new set of coordinates. The new basis is chosen based on the variance of the data—and the new coordinates are ordered by the amount of variance they account for. This ends up being useful for several reasons, one of which being that in high-dimensional data, random, uncorrelated fluctuations do not produce such large variance in the data as does correlated variation. In many situations, the correlations among the data points are of prime interest, either because they reveal an underlying mechanism or coordination, or because different underlying processes, which produce distinct sets of correlations, can be separated and revealed.

9.1.1. PCA for sorting of spikes

In order to isolate the spikes of individual neurons from multi-electrode data, it is common to run PCA on the voltage traces of the electrodes (Figure 9.1). In this context, the separate dimensions correspond to the values of voltage in different small time bins around the peak of a spike, as measured on separate electrodes. If, for example, 40 time-bins are used to quantify the voltage trace at each spike, then the number of dimensions would be 40 multiplied by the number of voltage traces. With such characterization, it is possible to extract multiple distinct "spike signatures" and identify more neurons than electrodes used. For example, the data shown in Figure 9.1B appear separable as two distinct clusters of traces on the single electrode shown.

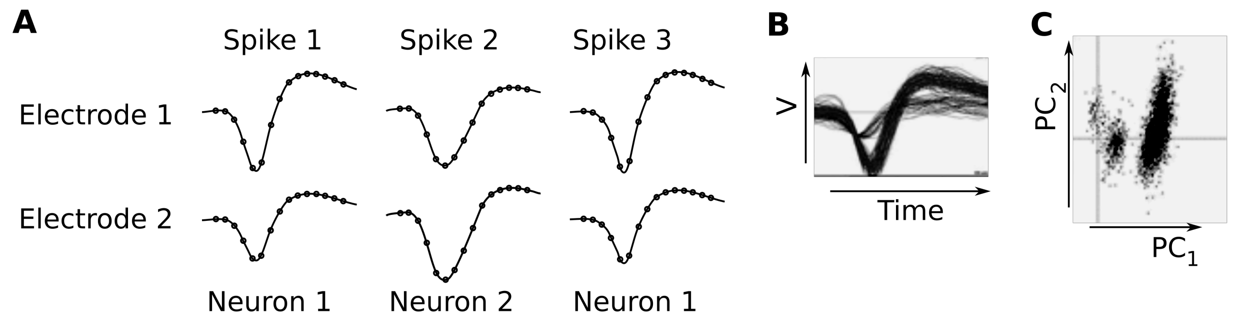

Figure 9.1. Separation of neural spikes by PCA. A. (Cartoon). When a spike is detected, the waveform is characterized by a number of values of the voltage (small circles), which together describe the characteristic shape produced on each electrode. Three spikes are depicted, each spike characterized by 15 values on each of two electrodes, which would produce a single point in 30-dimensions for each spike. Spikes 1 and 3 are similar because they arise from the same neuron, whereas spike 2 is significantly different, because it arises from a different neuron. **B.** Example traces of the multiple waveforms produced by several hundred spikes on a single electrode implanted in rat gustatory cortex. These traces group around two distinct forms, so most likely arise from two distinct neurons. Notice that the voltage trace, dependent on electric fields outside the neurons, can have a very different shape from the membrane potential of the neuron producing a spike. **C.** Each spike appears as a dot in high-dimensional space. The spikes separate into two clusters, representing the two putative neurons. Following PCA, the separation of the clusters is visible here when just the first two principal components of the data are plotted against each other. 40 data points per spike from a single electrode were used for this analysis. (Data provided by the laboratory of Don Katz, Brandeis University).

As noted above, PCA is helpful when there are correlations between the dimensions. For example, in multi-electrode recordings of electrical activity, any source of electric field, such as a single neuron, whose spike affects the voltage in more than one electrode, produces a correlation between the values recorded in the different electrodes. That is, whenever the neuron spikes, a certain shape of deflection of the voltage in one electrode will occur at the same time as a different shape of deflection of the voltage in another electrode. A different neuron would produce a different pair of simultaneous deflections in the two electrodes. Therefore, observation of a shape of deflection in one electrode correlates with the shape of deflection caused by the same cell in another electrode. Moreover, the values of a voltage-trace at successive time-points are highly correlated—if an electrode is at a point of high potential then a fraction of a millisecond later, the potential is likely to remain higher than average.

While one could imagine designing an algorithm to pick out the important features that differentiate the voltage traces of Figure 9.1A or 9.1B, these distinguishing features could vary from electrode to electrode and from cell to cell. The value of PCA in this context is that, once all traces are represented as individual points, it can allow us to see visually whether multiple clusters exist and use simple methods to separate the clusters.

9.1.2. PCA for analysis of firing rates

Once the activities of individual neurons are separated, correlations abound in the dynamics of extracted firing-rates. In this case, correlations between the activities of different cells can be caused by connections between the cells within a network, or by common "upstream" input. The interesting signal causing the correlations across dimensions can be more easily observed or extracted by choosing an appropriate linear combination of the measured data. For example, in a binary decision-making task (Section 6.5) one may want to add together the firing rates of cells responsive to one choice then subtract all the firing rates of cells responsive to the opposite choice, while ignoring the firing rates of cells with no choice preference. PCA can provide a systematic method for determining how much (and of what sign) each cell's firing rate should contribute to a particular "readout".

When analyzing neural activity, the initial, high number of dimensions is typically the number of neurons, with each dimension indicating the firing rate of a particular neuron. The variation of activity as a function of time would be a curve, or trajectory in the high-dimensional space as neural firing rates covary. If the covariation of activity lies predominantly along a 1D line, or a 2D plane, PCA would reveal this. For example, if the rates of all neurons deviated from their mean by different scaled amounts of the same function of time—*i. e.*, their rate changes were all in proportion to each other—the trajectory would be a straight line in firing-rate space. Even with noise added to each neuron's firing rate, PCA could reveal the straight line and produce a much less noisy estimate of the function of time coordinating the dynamics than that available from analysis of any neuron alone. In so doing, PCA may reveal the most important correlate of a stimulus or behavior to be found in the noisy neural activity. In Tutorial 9.1 we will practice such techniques.

9.1.3. PCA in practice

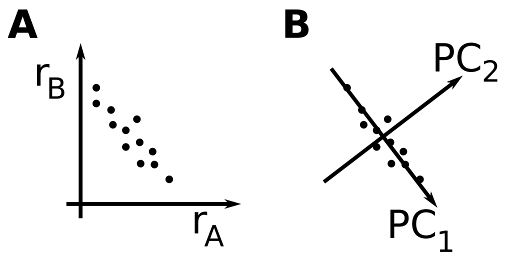

Figure 9.2. Selecting new axes with principal component analysis. A) Each data point corresponds to a specific measurement and comprises a pair of firing rates, r_A and r_B. **B)** The axis corresponding to the first principal component, PC_1, contains the most variance in the dataset. The second axis, PC_2, is perpendicular to the first.

The first step of PCA is the extraction of a new set of perpendicular axes sorted by the amount of variance of the data in the direction of each new axis. Figure 9.2 indicates how PCA would extract new axes from a pair of neurons whose firing rates are anti-correlated, meaning a higher rate of one neuron coincides with a lower rate of the other neuron and vice versa. In this example, the axis containing the most variance in the data, now labeled PC_1, is at 45° to the original axes and corresponds to the direction $r_A + r_B = constant$ (it could as easily have been the direction $-r_B - r_A = constant$). The second axis extracted, labeled PC_2, must be perpendicular to the first, so in this case is constrained to be the direction $r_A = r_B$.

Box 9.3. Projection of data onto an axis (or plane) reduces the dimensionality of data in the same way that a 3D image can be projected onto a 2D plane as a photograph—all depth information is lost. For example, if a set of (x, y) points are projected onto the x-axis then only the x-values remain and the y-values are lost.

The variance in the data accounted for by PC_1 is so much greater than that accounted for by PC_2 in this example, that a description of each data point by its first principal component alone—its value of PC_1—would not cause much loss of information. That is, if the data points in Figure 9.2B were moved to the nearest point on the PC_1 axis (such a movement is perpendicular to the axis and is called a projection onto the axis—see Figure 9.3) then little change in their position would ensue.

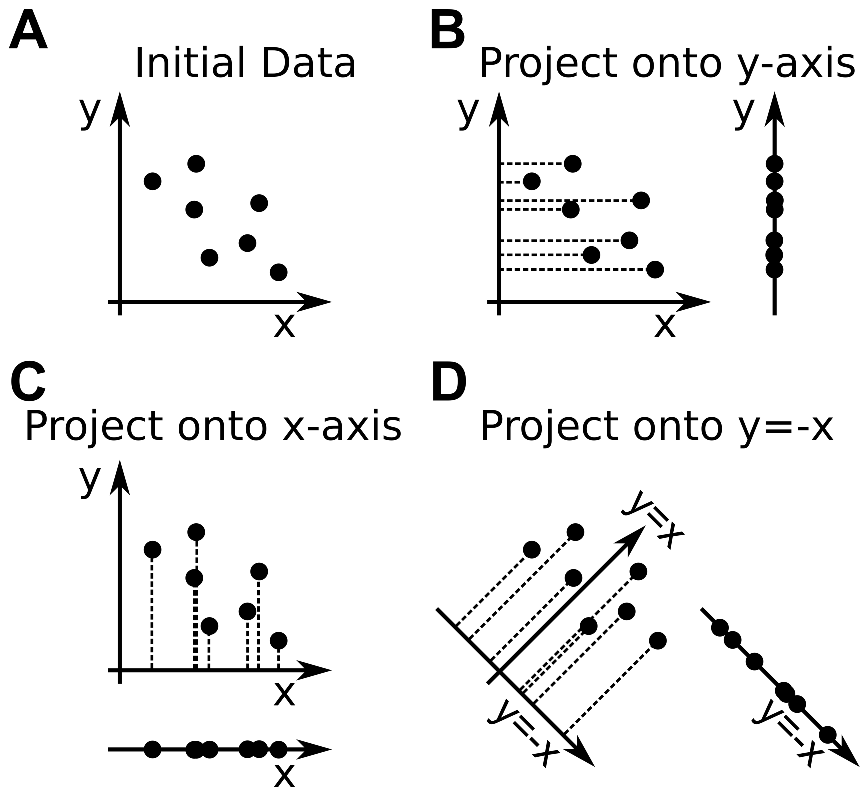

Figure 9.3. Projection of data. A. A set of data points in 2-dimensions can be projected to a single dimension in multiple ways. **B-D.** Projection of each data point (x_i, y_i) onto an axis is shown as a dashed line. The resulting set of projected data falls on a single line, so each data point is represented by a single number, its y-value (y_i) in B, its x-value (x_i) in C, or the scaled difference between the two $\left(\frac{x_i - y_i}{\sqrt{2}}\right)$ in D.

If the original data were three-dimensional (so had labels x, y and z, or equivalently, r_1, r_2 and r_3), it is possible that all the data points fall within a shape that looks like a tilted dinner plate. Such data would be closer to 2D as a plate is approximately a flat surface and a flat surface is a 2D plane. PCA would pick out two directions at right angles to each other within this plane, where the greatest spread in the data is found. If one plotted the data points on coordinates produced by these first two principal components one could see the data sitting essentially in an ellipse. The third principal component (in the direction of the thickness of the dinner plate, perpendicular to its plane) could most likely be ignored with little loss of information about the coordinates of any data point. Reducing the number of dimensions (or coordinates) used to describe data with minimal information loss is a form of data compression that can be used in some forms of data transfer.

Perhaps a key step in the use of PCA, or any other method for analysis of high-dimensional data, is the choice of what components of the data are used to produce new axes and extra dimensions for each data point, versus what components are used to produce new data points. For example, when firing rates of many neurons are measured as a function of time, the dimensionality could be the number of neurons. In this case, each time-point would provide a single data point. The set of data points for a single trial can then be joined together as a single curve, indicating the trajectory of neural activity as a function of time. PCA allows us to visualize such trajectories[5,22].

Alternatively, each time-point could add a new set of dimensions, such that the total dimensionality of the system becomes the number of neurons multiplied by the number of time bins. In this case, each trial would correspond to a single high-dimensional point (like each time window around a spike in Figure 9.1). Multiple trials are likely to produce a cluster of points, with trials from a different stimulus producing an alternative cluster (*cf*. Figure 9.1C). PCA is likely to extract the most significant distinctions between stimuli. These distinguishing features could then be mapped back to distinct signatures in the firing rate trajectories of the neurons.

Box 9.4. Covariance: The degree to which two series of data vary in the same direction as each other (positive covariance) versus in the opposite direction as each other (negative covariance). Zero covariance can arise if the two series vary independently of each other, or if either one or both of the series has no variation (zero variance).

Box 9.5. Eigenvectors: Eigenvectors are a set of orthonormal vectors—meaning perpendicular vectors of unit length—associated with a particular matrix. The eigenvectors are special, in that when multiplied by the matrix they are simply scaled rather than altered in direction.

Box 9.6. Eigenvalues: An eigenvalue corresponds to a particular eigenvector and is the amount by which that eigenvector is scaled when multiplied by the matrix.

9.1.4. The procedure of PCA

In this section, we describe the algorithmic steps that are required for PCA. In this description, each data point will correspond to the values of firing rates of each cell in a single time-bin, so the dimensionality is the number of neurons.

1) Center the original data by subtracting the mean in each dimension—*i. e.*, for every data point, subtract the mean (time-averaged) firing rate of the corresponding neuron, so we are analyzing deviations from the mean. The mean values should be stored if firing rates are to be recalculated. Subtraction of the mean facilitates rapid calculation of variances and covariances (recall that variance is calculated using deviations from the mean).

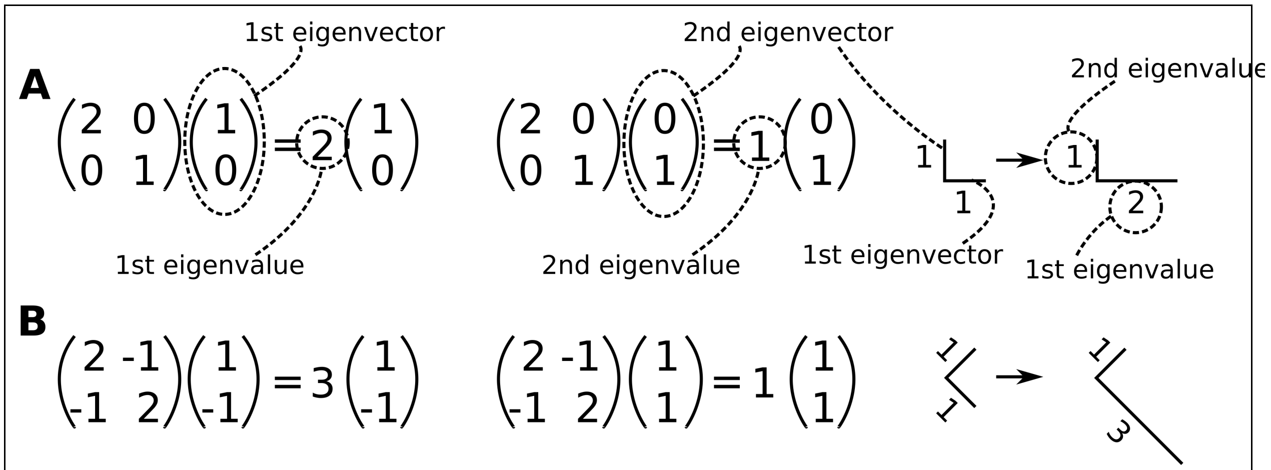

Figure 9.4. Eigenvectors and eigenvalues. A. The matrix $\begin{pmatrix}2 & 0\\0 & 1\end{pmatrix}$ has an eigenvector $\begin{pmatrix}1\\0\end{pmatrix}$ directed along the x-axis with an eigenvalue of 2, because multiplication of that eigenvector by the matrix simply scales the vector by a factor of 2. The second eigenvector, $\begin{pmatrix}0\\1\end{pmatrix}$, is perpendicular to the first, directed along the y-axis, and has an eigenvalue of 1. Notice that the eigenvectors could equally have been chosen to be in the opposite direction, *e. g.*, $\begin{pmatrix}-1\\0\end{pmatrix}$ is an eigenvector, and they would still be perpendicular with the same eigenvalue. **B.** The matrix $\begin{pmatrix}2 & -1\\-1 & 2\end{pmatrix}$ has eigenvectors rotated by 45°. Since eigenvectors should be of unit length, the actual eigenvectors are $\frac{1}{\sqrt{2}}\begin{pmatrix}1\\-1\end{pmatrix}$ and $\frac{1}{\sqrt{2}}\begin{pmatrix}1\\1\end{pmatrix}$, with eigenvalues of 3 and 1 respectively. Notice that when the matrix multiplies a vector that is not an eigenvector, the result is not a scalar multiple. *e. g.*, $\begin{pmatrix}2 & -1\\-1 & 2\end{pmatrix}\begin{pmatrix}1\\0\end{pmatrix} = \begin{pmatrix}2\\-1\end{pmatrix}$ means a vector along the x-axis is rotated clockwise as well as scaled by the matrix in B.

2) Calculate the covariance matrix of the centered data (this can also be done without step-1 since the covariance is mean subtracted). The covariance matrix will be a symmetric $P \times P$ matrix if there are P neurons. Each entry in the covariance matrix is calculated for two neurons, by taking the product of their firing rates together in each

time-bin then calculating the mean of these values across time bins. The diagonal of the matrix is simply the variance in firing rate of the corresponding neuron.

3) Find the eigenvectors of the covariance matrix with their eigenvalues. (An eigenvector is a particular direction or axis, such that a vector in this direction, when multiplied by the matrix, is simply scaled in the same direction. The amount of scaling is the eigenvalue—see Figure 9.4. Many coding languages can produce eigenvectors and eigenvalues of matrices using a single command—it is "`eig`" in Matlab.)

4) Sort, then order, the eigenvectors from the one with the largest eigenvalue to the one with the smallest eigenvalue (all eigenvalues are non-negative real numbers, since the covariance matrix is real and symmetric). The sum of the eigenvalues is the sum of the variances of firing rates of all neurons. When each eigenvalue is divided by this sum, the result yields the fraction of the total variance of neural activity due to coherent variation of the system in the direction of the corresponding eigenvector.

The eigenvectors, each of which defines a direction providing a new axis, are perpendicular when created, so we have generated a new set of perpendicular axes for the data—*i. e.*, a new basis. The axes are ordered according to the directions with highest to lowest variance in the original data, which is the order of the principal components.

9.2. Tutorial 9.1. Principal component analysis of firing rate trajectories

In this tutorial, you will first generate firing rate trajectories of 50 neurons. Each neuron's firing rate is produced by a weighted combination of two input signals, mixed in with a lot of noise. You will then use PCA to produce less noisy firing rates for each cell (de-noising) and to extract the input signals from the noisy set of firing rates.

1. a) Define two vectors to represent time-dependent oscillating inputs, I_A and I_B, such that $I_A = A\sin(2\pi ft)$ and $I_B = B\cos(2\pi ft)$, with frequency $f = 0.5$Hz, and amplitudes $A = 20$ and $B = 10$. You can use time-steps of 1ms and a duration of 10s.

   b) Set up a matrix to contain the firing rates of 50 neurons across this time-span. Generate the firing rate of each neuron as a column in the matrix, with each row representing a separate time-point. The time-dependent firing rate of a neuron, labeled i, is given by:
$$r_i(t) = 100 + W_i^{(0)} I_0 + W_i^{(A)} I_A + W_i^{(B)} I_B + \sigma \cdot \eta_i(t),$$
   where $I_0 = 50$, the static input weights, $W_i^{(0)}$, $W_i^{(A)}$, and $W_i^{(B)}$, are each independent numbers selected separately, once for each neuron, from the Normal distribution with unit standard deviation and zero mean (defined as $N(0,1)$ via `randn()` in Matlab). σ=10 scales the noise, and $\eta_i(t)$ is a series of Normally distributed random variables, $N(0,1)$, selected independently at each time-point for each neuron.

c) Carry out principal component analysis on the rate matrix, using (in Matlab) the command:

```
[COEFF,SCORE,LATENT,TSQUARED,EXPLAINED,MU] = pca(rate);
```

Note that the `pca` function in Matlab requires the data in each dimension to be stored as a set of column vectors, with each column being a different dimension. In Matlab you can use "`help pca`" to find the meanings of the outputs. The important ones for this tutorial are:
`COEFF`, which defines the new basis. Each column vector in `COEFF` is a principal component, which is an eigenvector of the covariance matrix. The vector has unit length with elements given by the relative contribution of each neuron (in this example) to the principal component.
`SCORE`, contains the representation of the centered data in the new basis. Each row of score which corresponds to a single data point, with each entry its value in the new coordinate system, that is, its projection on each successive principal component.
EXPLAINED, which is a vector containing the percentage of variance explained by each principal component.
MU, which is a vector containing the mean firing rates of each neuron, as needed to recreate the original rates.

d) Plot the first column of coefficients, `COEFF`, (corresponding to the contribution of each neuron to the first principle component) against the vector of input weights, $W_i^{(A)}$, and, on a separate subplot, the second column of coefficients (corresponding to contribution of each neuron to the second principal component) against the vector of input weights $W_i^{(B)}$. Does the sign of the slope of any observed trends matter?

e) Plot the variable, `EXPLAINED`, to ensure the bulk of the variability in the data is contained in the first two principal components and calculate that fraction explained by summing its first two values.

f) To de-noise the data, you will produce a new matrix of firing rates by multiplying the first two columns of `SCORE` by the first two rows of `COEFF'` (the transpose of `COEFF`) and adding to all entries of each column produced in this manner, the value of the corresponding entry of `MU` (the mean rate of that neuron).
In this step, we are assuming that once in the coordinates of the principle components, any changes in rates on principal component axes 3 or higher correspond to noise fluctuations that can be ignored. Therefore, we are recreating the firing rates of the original neurons just using the projection of each neuron's firing rate on the axes of the first two principal components.

g) Compare the behavior of the de-noised data with the original data by plotting (on separate subplots) the original rates of two neurons as a function of time, then the corresponding columns of the new matrix to reveal the de-noised rates as a function of time.

h) Select two neurons and plot the rate of one against the other, both using the original noisy rates, and, in a separate subplot, using the de-noised rates.

i) Plot separately the first and second columns of the matrix, `SCORE`, to show the time-dependence of the system's first and second principal components.

Be sure to comment on all of your observations and relate them to the initial set of inputs.

2. Repeat question 1, but in part b) define $I_A = A\sin(2\pi f_A t)$ and $I_B = B\cos(2\pi f_B t)$, where $f_A = 1$Hz and f_B=0.5Hz, keeping all other parameters the same.

3. Repeat question 1), but define one input as a slowly ramping current, $I_A = At$ (from $t = 0$ to 10s), and the other as a transient signal during the ramp, $I_B = B\sin(2\pi f t)$ for 4s $< t < 5$s, otherwise $I_B = 0$.