Lab 8: Continuous decoders

Introduction

In this lab, we will begin building towards implementing a number of real-time decoding algorithms that have been reported in literature, using published research as reference. All algorithms presented in the next lab are based heavily on linear regression, so we start with simple linear regression and then move on to fancier things.

Unlike previous labs, this lab will provide relatively little guidance on code structure, leaving you to determine the most appropriate form of implementation. Note that many of these algorithms can be implemented very efficiently and concisely using matrix notation provided in lecture. All code should be written from the ground up.

Software

This lab must be completed using MATLAB.

Part 0) Process data

For all parts of this lab, we will be using the same set of data from a reach task. The data needs to be separated into training and test sets for subsequent parts of the lab.

1. Load contdata95.mat. Columns of X contain X position, Y position, X velocity, and Y velocity at 31413 time points. Columns of Y contain firing rates of 95 recorded units.
2. The data is in 100 ms bins. For each neuron, create 9 additional features that represent the neural data delayed by 100 ms, 200 ms and so on such that you have 950 total features.
3. Add a column of ones to the neural data. This is necessary for calculating an intercept term in the regression models. You should now have 951 "features" to predict from.
4. Designate the first half of X and Y as training data and the second half as test data.

Part 1) Linear regression

Decoders based on linear regression are a popular real-time decoding algorithm because of their simplicity and ease of interpretation. The objective function for linear regression is

$$\min_{\vec{w},b}\left[\frac{1}{n}\sum_{i=1}^{n}(\vec{w}^T\vec{x}_i + b - y_i)^2\right]$$

where $\vec{x}_i$ is a vector of features (e.g., firing rates of different neurons) for observation i, y_i is the measured outcome of observation i (e.g., hand position or velocity), and w_i and b are estimated parameters of the linear model.

Using the equations provided in lecture, fit a linear regression model to the training data. Then report the following findings:

1. Create a plot showing x and y position for at least three movements with the actual movement plotted on top of the predicted movement. Include a legend.
2. Mean squared error of predictions on the test data. Report a value for each output parameter.
3. Correlation of predictions to actual motion. Report a value for each output parameter.
4. Discuss the advantages and disadvantages of a linear regression decoder. Be sure to include discussion of assumptions inherent to linear regression.

Note: It may be helpful to include mean squared error and correlation values in a table.

Part 2) Ridge regression

Ridge regression is just like linear regression, except it includes a normalization term λ that penalizes large values in $\vec{w}$. The objective function of ridge regression is

$$\min_{\vec{w},b}\left[\frac{1}{n}\sum_{i=1}^{n}(\vec{w}^T\vec{x}_i + b - y_i)^2 + \lambda\left|\left|\vec{w}\right|\right|_2^2\right]$$

where $\left|\left|w\right|\right|_2$ is the L_2 norm of w. By penalizing large values in $\vec{w}$, ridge regression prevents the algorithm from overfitting to a few sporadically firing cells. (Note: Do not use the ridge() function for this part.)

1. Using the training data, perform a parameter sweep of λ, for 15 equally-spaced from 0 to 0.1. That is, fit a ridge regression model for each value of λ and report which value results in the most accurate predictions. Provide 1 or 2 plots that show your parameter sweep, indicating which λ is most accurate. Note: Lecture notes say $\beta = (X^TX + \lambda I)X^TY$ (without N), whereas your regression equation in code should be $\beta = (X^TX + N\lambda I)X^TY$ where N is the total number of training datapoints.

2. Using the best value of λ, apply the fitted model to the test data. Report mean squared error and correlation of each output parameter.

3. Discuss how this compares to regular linear regression above.

Guidelines for Lab Report (on Labs 7, 8, and 9 together)

Introduction: The introduction should be one paragraph long summarizing the motivation for developing the tools used in this lab and what they can be used for, along with a brief summary of everything you will show in this lab report.

Methods: From Lab 8, there should be methods paragraphs (and diagrams where necessary) on:

1. Assumptions of the algorithms used
2. How the algorithms were implemented

Include the code as an Appendix to your report. Cite sources for any values used in your models.

Results: You should include the following in your Results:

1. The outcome of each algorithm (correlation and mean squared error, along with plot of parameter search in Part 2).
2. The limitations of each algorithm.
3. How the algorithms compare to each other.

Include all figures produced by MATLAB that could help explain and illustrate your findings.

Discussion: Should be ~2 paragraphs long describing what you could use these tools for in the future.

This report will be combined with Labs 7 and 9 to create one cohesive report. The report (not including Appendix) should be no longer than 6 pages. Use 12 pt. font and 1.15-1.5 line spacing. If your text is over the 6-page limit with figures, you can move your figures to an appendix section that goes beyond the 6-page limit. Please upload your report to Canvas.