-First train a simple feedforward network/linear regression that looks at past 5-10 datapoints to predict position, then make network as complex as you want

-we need to think about how to regularise the data (timestamps at the same time), average of timestamps in between

-maybe sufficient to average all and have it coarse to reduce data without increasing complexity of learning task

-rate of xy position is 39 datapoints p/s, plot data see how much mouse moves in 1-5 sec, if most of the time mouse is standing still then maybe decrease resolution with which we’re feeding data. Only have one xy coordinate per second-ish, so reduce dataset (average over entire second if mouse doesn’t move much) exploratory data analysis

-what does our algorithm have difficulties with?

-less movement might negatively impact our algorithm if it increases the loss

-think about time resolution

-when you have a real world data set, data analysis data cleaning etc is the biggest part, so learning alg has best starting point.

-figure out whether preprocessing is worth it or do a generic learning task with already processed data

-is underestimated a lot

-super useful for jobs or internship opportunities

-isolate xy coordinate data

-PCA on time series that take into account temporal relationships

-stick to one experiment, run basic model (first 80% and use remaining 20% to predict)

-define basic thing we definitely want to achieve and later define other goals we can do if we have time

-bootstrap learning, transfer learning

-many to many

-google collab

<https://deeplearning.cs.cmu.edu/S21/document/recitation/Recitation_6_RNN_Basics.pdf>