The idea is that the history of the calcium activity affects the probability of observing a spike. Keeping this in mind, we implemented an LSTM that takes the previous history of two seconds as the input. The RNN has 64 units, that takes inputs in 10 chunks, each chunk containing 20 time points. The data was fed in batches of size 128 and the network was run for 10 epochs for each of the 5 folds in cross validation. The dropout was 0.8. The observed accuracy was 78.2% and 48.6% were accurately predicted by the model. However, the model had massive false positives. The following plot shows the spike count (in a broad window of 100ms) for the model (red) compared against the actual spike count (blue). The cost function was cross entropy, penalized if the model used a large number of predicted spikes.

<Image1>

The reason for why the model didn’t work satisfactorily can be inferred by zooming into the 1st second:

<Image 2>

The neural network puts spikes all across the time series, but the actual number of predicted spikes is small compared to the overall length of the time series (not visible in the plot). This ensures that the accuracy is not abysmal. Also, since we accepted a spike predicted one sample (10ms) before or after the actual spike as a correct prediction in this model, the percentage of predicted spikes is also not small. Thus, the model does optimize accuracy and percentage of predicted spikes, but it does so by putting a small number of spikes all over the time series – which is not biologically desirable.

Future directions:

1. Explicitly involving the derivatives and their history in the RNN.
2. More sensible cost function optimized to predict higher percentage of spike match, and not just overall accuracy.