Deep Queue-Learning: A Quest to Optimize Office Hours

Avoy Datta, Dian Ang Yap, Zheng Yan
CS 229 | Autumn 2018

CS 221

CS 229

Introduction

- OHs often suffer from **overcrowding and long wait times**, stressing both students and instructors.
- If we could accurately predict the expected workload at a given OH, TAs can be better allocated.
- QueueStatus, Carta, and course syllabi provide a wealth of information that can be used.
- We trained a **neural network model that predicts student load influx (expected serve time * # sign-ups)** at OH on an hourly basis, for any course.
- With these predictions, we now **optimize TA scheduling** given **realistic constraints**.

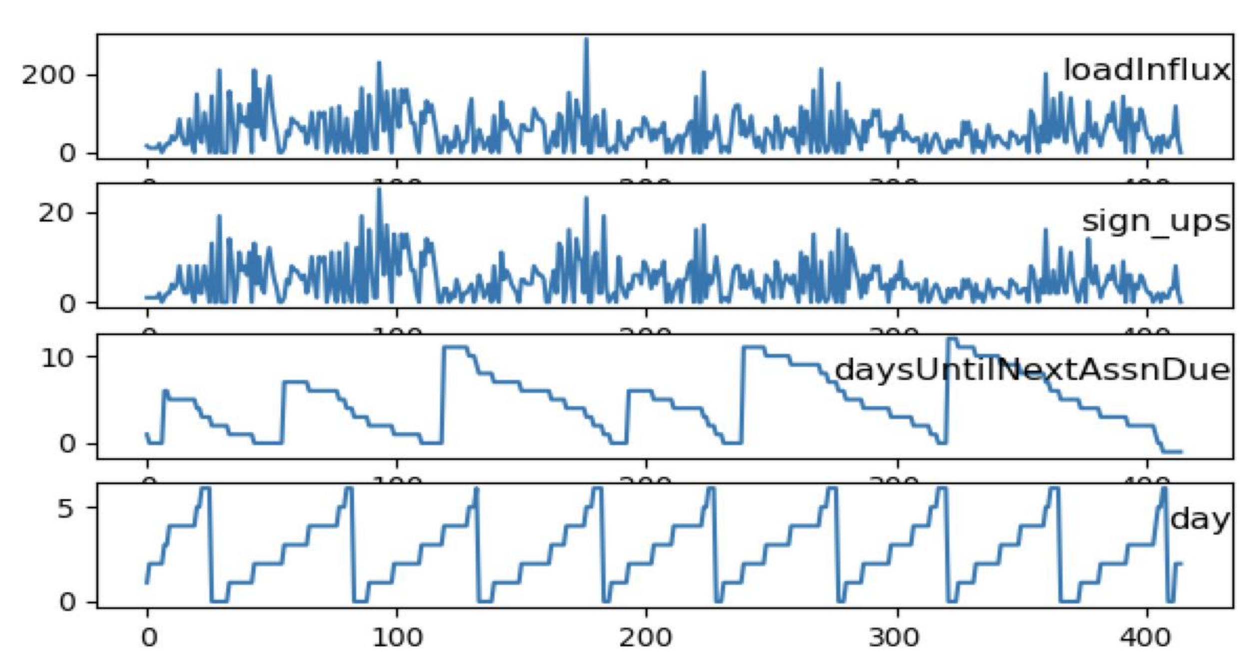
Class Statistics

Table 1: Statistics for sample of classes (4/8 shown)

Class	Quarter & Year	#OH-Active TAs	Total # Students	Total OH Hours	Total Served	Total Load Influx
CS107	Spring 2017	13	184	415	1722	21873.09
CS161	Spring 2017	6	93	204	875	15380.68
CS110	Spring 2018	20	187	223	1749	35459.1
CS229	Autumn 2018	17	634	369	1390	31733.7

Features and Preliminary Statistics

- Load influx is significantly and positively correlated with: **Week number** (r = 0.07) and **Number of servers** (r = 0.32)
- Significantly and negatively correlated with: **Days left until assignment due** (r= -0.08), **Hour of day** (r = -0.10), **Weekday** (r=-0.09), **Days until next exam** (r = -0.06)



Methodology

- We defined a new loss (Shrug loss) and used smoothing on labelled data to reduce penalty on outliers.

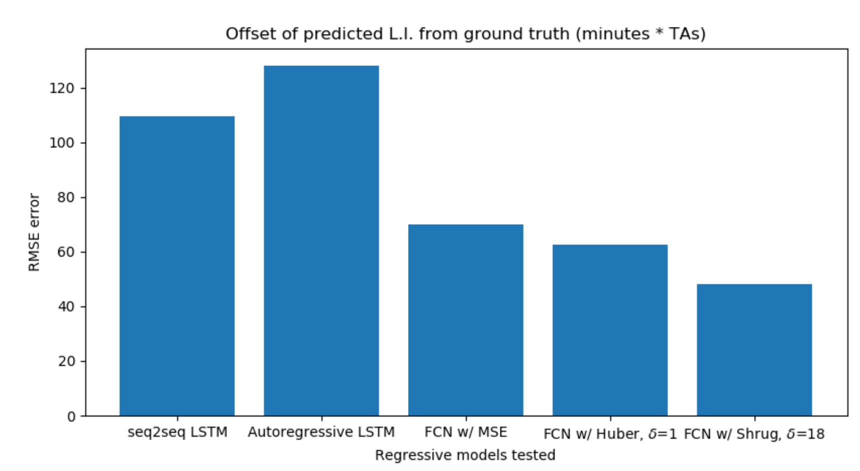
$$\mathcal{L}_\delta = \begin{cases} \frac{1}{2}(y-\hat{y})^2, & \text{if } |y-\hat{y}| \le \delta \\ \sqrt{\delta(y-\hat{y}) - \frac{1}{2}\delta} & \text{otherwise} \end{cases}$$

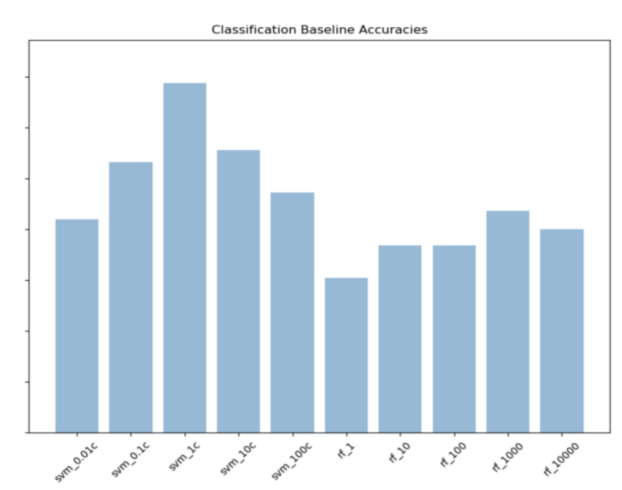
- To reduce contributions by outliers, we smooth out the data through convolution with a *Hann* window:

$$w(n) = \frac{1}{2}\left(1 - \cos\left(\frac{2\pi n}{N-1}\right)\right)$$

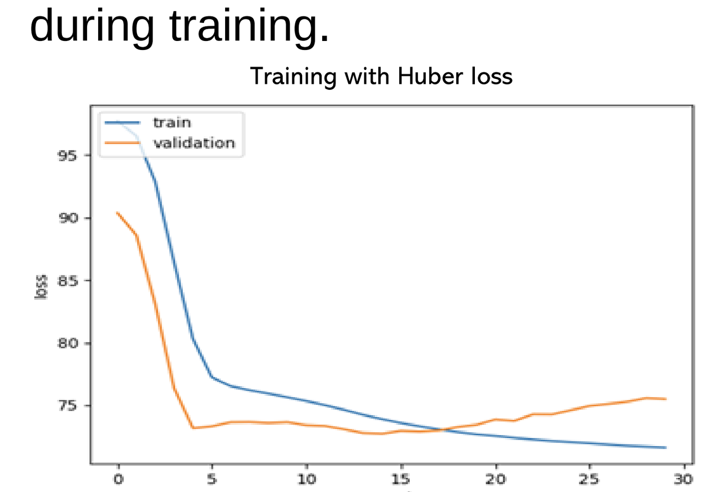
Experiments

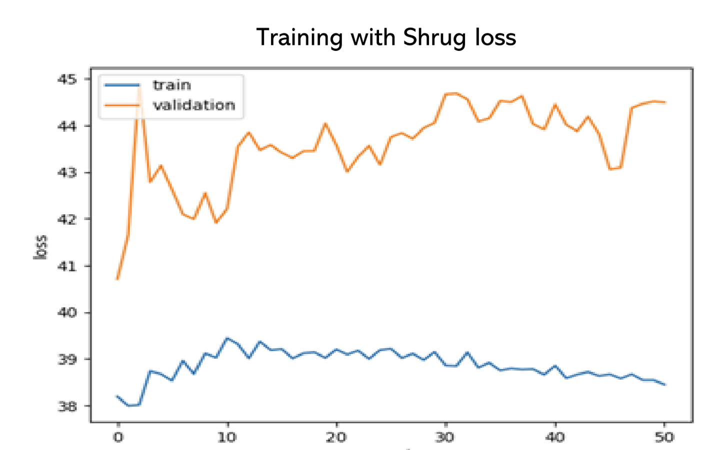
- Both classification (SVMs, Random Forest) and regression (fully connected nets, LSTMs) were experimented for predictions.





- Shrug yields significantly lower RMSE on test set, but poorer convergence during training.





Scheduler

- We use **Gibbs Sampling** to assign TAs to each individual time slot.
- The Gibbs sampler optimizes:

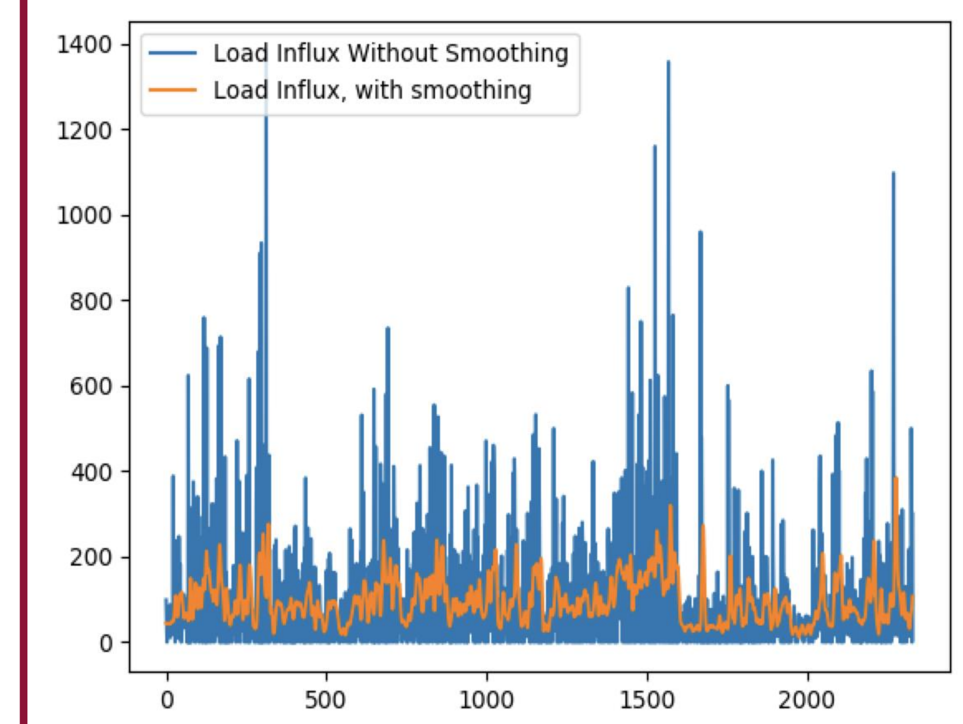
$$P\left(X_{ij} = x\right) = \boldsymbol{T_{assigned}} \cdot \boldsymbol{T_{predicted}} \mid X_{ij} = x$$

$\boldsymbol{T_{assigned}} \cdot \boldsymbol{T_{predicted}}$ *measures the* ***cosine similarity*** *between the number of Tas assigned each office hour and the predicted loads*

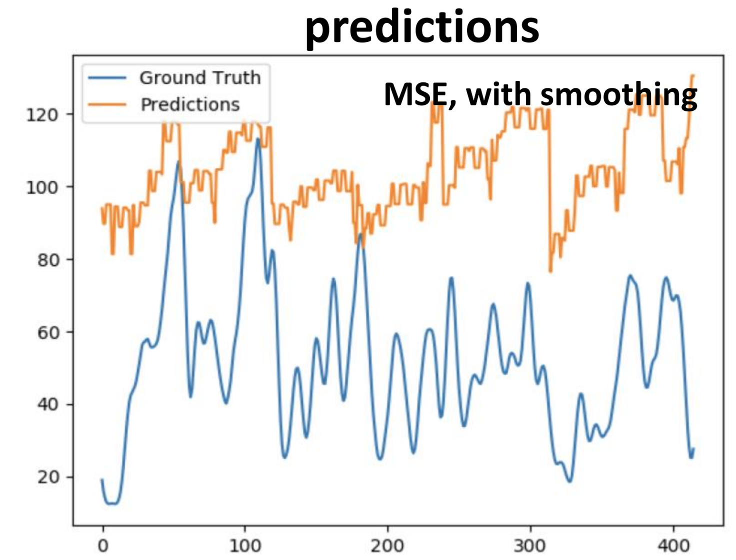
- Weight of sampling is proportional to *increase* in cosine similarity of the full assignment for each value assigned.

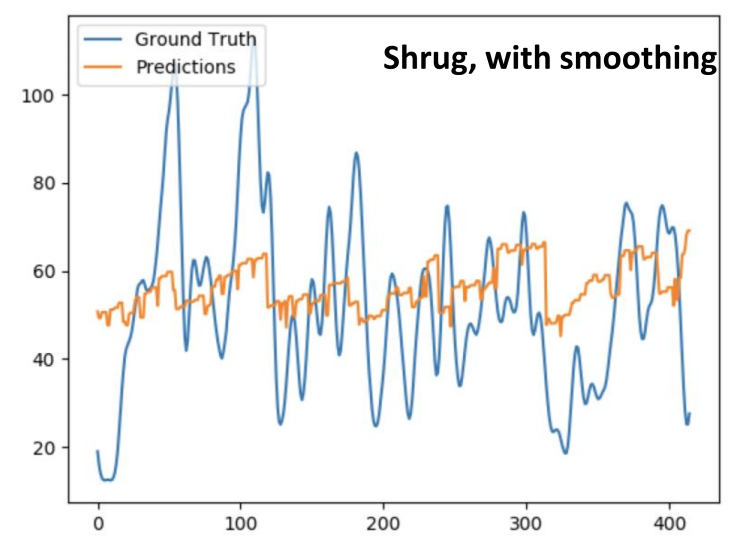
Results

Visualization of smoothed y-labels

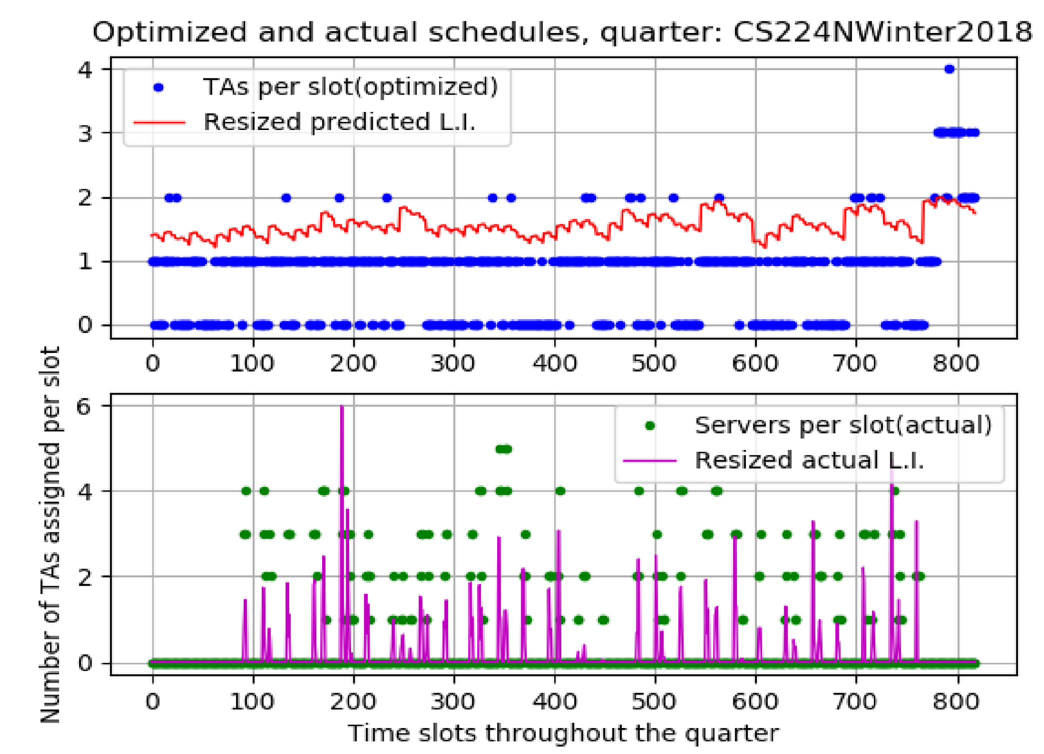


Shrug loss reduces offset of predictions





- Smoothing reduces spikes in erratic data.
- Less effective in predicting actual magnitude.



Loss Functions w/ FCN	RMSE (Load Influx)
MSE	69.89
MAE	62.65
Huber (δ = 1)	62.61
Shrug (δ = 18)	**48.0**

Cosine similarity, actual schedule	Cosine similarity, optimized schedule
0.794	0.789

Summary

- Using data scraped off of Stanford course resources, a fully connected NN, and Gibbs sampling, we have come up with a system that schedules TA hours (within realistic constraints) that appears to correlate well with student demand.
- Major challenges for inference: figuring out a model that balanced bias with variance and coming up with a loss that didn't penalize outliers excessively.
- This model can serve as a recommender system for office hours for newly introduced courses. We tested it on one quarter of a course not used in the train set and found correlation between assigned hours and predicted influx were similar to actual load influx and server correlation.