Deep Queue-Learning: A Quest to Optimize Office Hours

Avoy Datta, Dian Ang Yap, Zheng Yan

CS 221

CS 229

Note: This project is shared between CS221 and CS229. For CS221, we focus on TA scheduling.

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**.

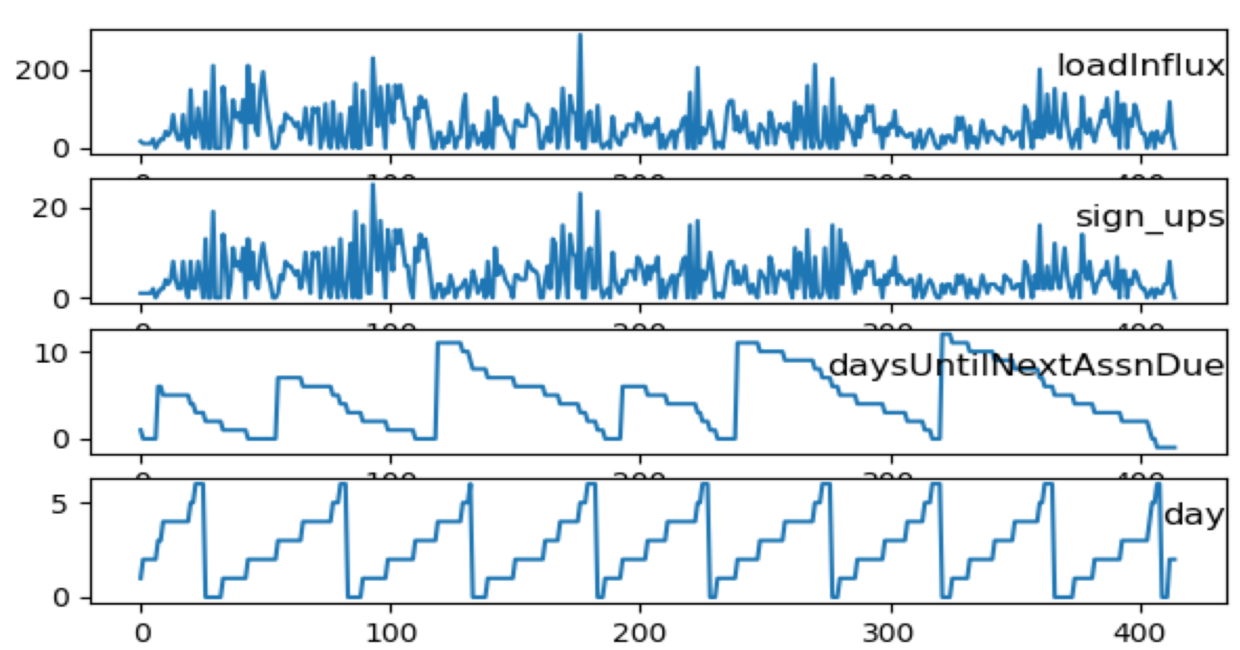
Class Statistics

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

Collect Queue Data → Predict Load Influx → Optimize TA Scheduling

Training Datasets: **CS107** (Spr/Aut 2017, Aut/Win 2018), **CS161** (Spr/Aut 2017), **CS110** (Spr/Aut 2018), **CS229** (Aut 2018)
Test Dataset: **CS107 (**Spr 2018)

Predicting Student Load Influx

- Trained and compared different models with deep fully-connected NN, univariate LSTMs and multivariate LSTMs.

Model for regression	RMSE (Load Influx)
FCN (hinge/MSE)	161 / **119**
Univariate LSTM	137
Autoregressive LSTM	128

Optimizing TA Scheduling Using Predicted Load Influx

- Having predicted the expected load influx for each individual office hour in the quarter, we use Bayesian inference with **Gibbs Sampling** to assign TAs to each individual time slot.
- Treating each TA as a variable, the Gibbs sampler assigns a *fixed*-length list of time slots to each TA through the quarter. Each time slot assignment is weighted according to:

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

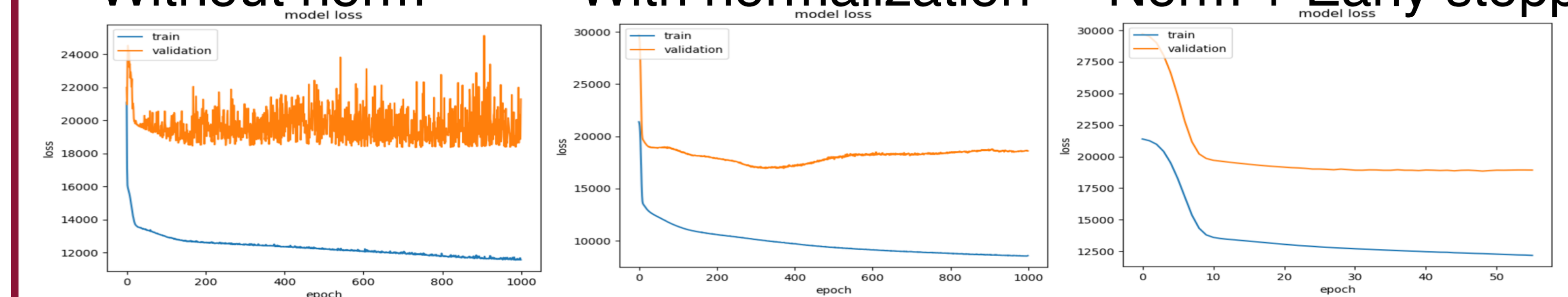
$\boldsymbol{T}_{assigned} \cdot \boldsymbol{T}_{predicted}$ *measures the* ***cosine similarity*** *between the number of Tas assigned each office hour and the predicted loads*

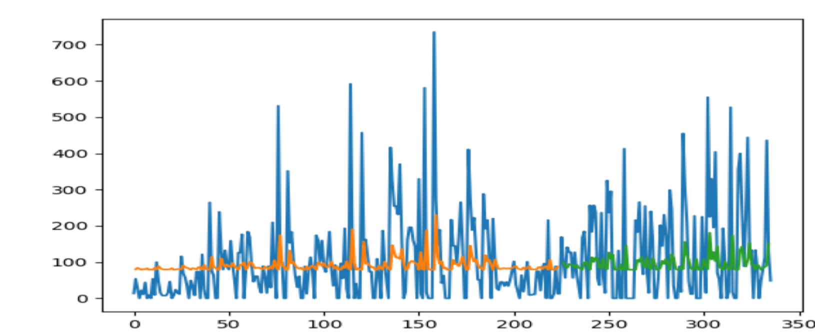
- We also factor in constraints such as a maximum TA workload per day and other commitments for TAs, and we reward continuity of office hours for a TA.
- Due to the high tendency for students to change their behavior according to assigned OH schedules, finding metrics to compare our results to the status quo (without real life implementation) is difficult.

Results

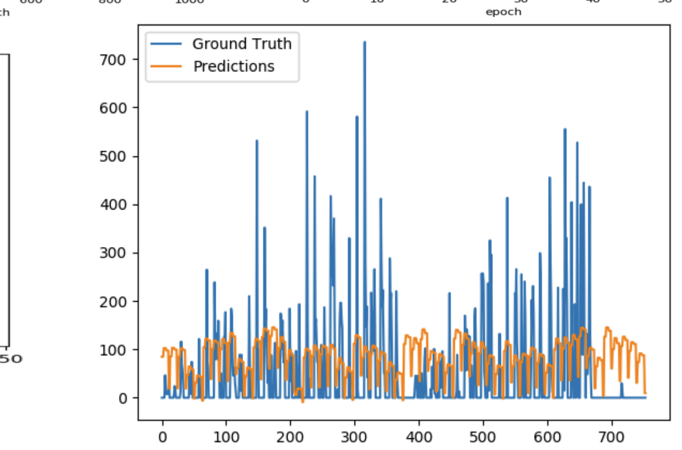
- We used normalization and early stopping to prevent overfitting in our NN models.

Without norm / With normalization / Norm + Early stopping



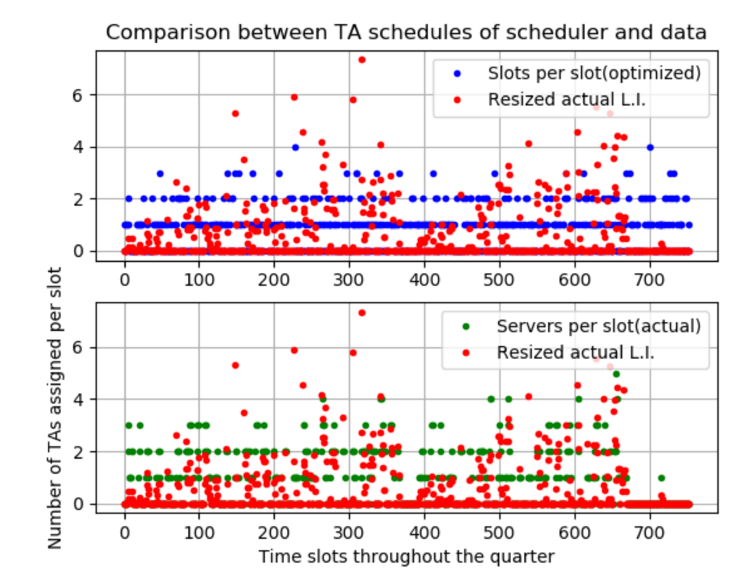


Initial mode (Univariate LSTM) predicts load influx spikes. (Blue- actual, orange – train predictions, green– validation predictions). Y-axis is the load influx, X-axis represents OH timeslots in chronological order.



Final model (Fully connected NN) predicts load influx spikes. (Blue- actual, orange – test predictions). Y-axis is the load influx, X-axis represents OH timeslots in chronological order. We see that predictions roughly line up with real spikes in student demand. (besides in the last week, when OH was not offered in the actual quarter).

CS107 Spring 2018 (Test set)



Comparison of suggested TA schedule and actual TA schedule. We see that our recommendations roughly line up with real spikes in student demand

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 included computational constraints, overfitting, applying real-life TA constraints, and deciding criterion of "optimality" for assignments.
- Over the next week, we will extend this to more general testing data, including courses we have not seen before (ex. CS224N). We will also identify and implement metrics to estimate the efficacy of our model compared to the status quo.