TRANSFOR19 Missing Speeds "Neural" Network

Carlin Liao, Venktesh Pandey, Tengkuo Zhu, William Alexander, Rahul Patel, Manoj Gedela
SPARTA Lab
The University of Texas at Austin

December 10, 2018

1 Introduction

This model uses the physical characteristics of the road grid to inspire the design of a predictive neural network and give its internal structure explanatory power.

2 Exploration and Data Processing

In our initial exploration of the dataset we restricted ourselves to only examining the `gps_20161201` dataset for potential features to include in our model. Slicing the datapoints to include only those in the region we're asked to estimate speeds for, we attempted to replicate the speeds given in the two prediction CSVs by pairing temporally consecutive timestamps in each order ID and calculating their distance and speed per the formulas given by the organizing committee. (These pairs of GPS coordinates and timestamps, which we will refer to as "deltas" or "steps" from now on, form the basis of our analysis.)

On grouping deltas' speeds per five minute intervals and taking their average, we noticed that our averages were closest to the given speeds when (1) our northbound speed averages included points not changing in latitude and (2) our southbound speeds did not. We also noticed that our averaging method seemed to slightly overestimate northbound speeds and slightly underestimate southbound speeds compared to the given predictions CSVs, so we introduced correction terms of $+1.3$ and -2.6 respectively to our averaging method.

Next, we attempted to glean the dataset for information that could give insight into the speed in the missing section, based on the priciple that the traffic that determines the northbound and southbound speeds in the area of study must travel through somewhere else in the network, and thus speeds on neighboring links should be a strong predictor of the average speed in the missing section. Isolating the road segments immediately north (N and NN), south (S and SS), east (E), and west (W) of the missing section (the latter of which we will refer to as the "area of study", "bounding box" or "bbox" from now on), we grouped movements regarding these areas per every five minutes (to match the predictions we are asked to make) to see what trajectories represented by the deltas were numerous enough to significantly contribute to our estimate. The areas are highlighted with grey boxes in Figure 1. Note that areas N and S end just before their each intersection.

Figure 1: Labeled segments near the area of study.

In our analysis, we found insufficient turning movements to inform most five-minute intervals so restricted our analysis to the movements depicted by the arrows in Figure 1. The red and magenta arrows represented close to 100% of deltas starting or ending in areas N and S, the road segments through which almost all vehicles traveling through the area of study pass through. Deltas starting and ending in E as well as in W also had counts high enough when batched per 5-minute intervals to be included in our analysis. Thus, we decided to isolate 5-minute average speeds for the following 14 movements identified by their start and end areas to include in our analysis:

- Included in the northbound speed prediction
  - `n_n_nb`: N to N northbound
  - `n_nn`: N to NN
  - `s_n`: S to N
  - `s_s_nb`: S to S northbound
  - `ss_s`: SS to S
- Included in the southbound speed prediction
  - `n_n_sb`: N to N southbound
  - `nn_n`: NN to N
  - `n_s`: N to S
  - `s_s_sb`: S to S southbound
  - `s_ss`: S to SS
- Included in both predictions
  - `w_w_eb`: W to W eastbound
  - `w_w_wb`: W to W westbound
  - `e_e_eb`: E to E eastbound
  - `e_e_wb`: E to E westbound

Note that the east-west deltas are included in both northbound and southbound predictions because vehicles traveling north and south can turn on and off both east and westbound links, as well as the fact that small segments of the east-west links are included in the area of study and thus would have an effect on the average.

For deltas that we could not determine a trajectory for (e.g. if a delta in area N did not change latitude over the two timesteps), we allocated half of nonmoving deltas in a five-minute interval to one direction and the other half to the other direction and reduced their averages accordingly. This way we avoid missing stop-and-go traffic speeds over a five-minute interval.

(For more details on this analysis see `1_exploration_2.ipynb`.)

Given Didi GPS data from October and November of 2016, we identified each non-holiday Monday, Tuesday, Wednesday, and Thursday to use in model training and validation because those weekdays would be most similar to our study date, Thursday December 1st. For each day, we removed all GPS points in the area of study and calculated the average speeds per five minute intervals for the 15 trajectories listed above on the resulting deltas. We also created a comparison vector for each day by averaging the northbound and southbound speeds in the bounding box per five minute intervals as described earlier.

Each day of GPS coordinates took about a minute to process on this i5-8600 desktop computer.

(For more details on data processing see `2b_processing.ipynb`.)

3 Model and Results

The goal of our model was to use the basic structure of a neural network in order to generate better and better speed predictions using the input data we identified. Unlike the typical neural network where the middle layers are typically a "black box", we designed our network such that each layer has at least some physical meaning.

Figure 4 depicts our network structure. Using the average speeds of all 15 movements as described earlier, we create better and better speed predictions for the area inside the bounding box. Using solely the northbound (magenta) or southbound (red) movements as described earlier, we make our first speed prediction. Introducing the east- and westbound movements (green) in the second layer of the network, we improve upon our first prediction with more information to make a second guess. Finally, we include the speed prediction from the *oppositely* bound direction as well as our prediction from the prior timestep to introduce some continuity in speed across time and direction to produce our final speed predictions for both north and southbound directions.

Each hidden layer is essentially a nonlinear "averaging" of different speeds from different sources that, combined with the loss function comparing our predicted speeds to the observed speeds, produce better and better averages the deeper the network we go. Instead of having to translate speeds on adjacent links into vehicle flows using an estimated volume-delay function before translating the resulting flows back into speeds, we instead estimate the resulting speed directly using the hidden layers as "universal function approximators" that can estimate the nonlinear relationship between adjacent link speeds and the speeds in our area of study.

In addition to using speed predictions from the prior timestep in our model, we also use the 15 movement speeds from the prior timestep because traversing the interval could take more than five minutes, so there is some information to be gained from knowing

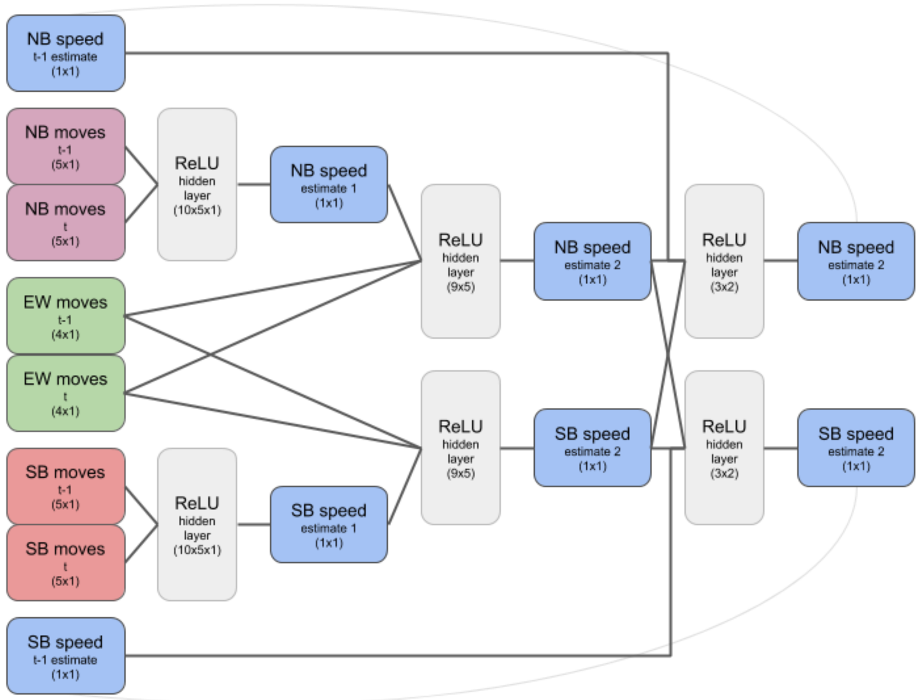

Figure 4: Neural network structure.

how fast vehicles were traveling around the area five minutes beforehand.

We also want to emphasize that our model does not use information from the "future"; that is to say that the speed prediction for time t is only dependent on information from time t, $t-1$, and so on. This is both for computational efficiency as well as to simulate real-world conditions where a real-time, rolling prediction is a more realistic use case.

Consequently, we trained our model in two ways. First, we had our model make predictions over an entire day then calculated the RMSE and updated weights accordingly. This is our "static" model. Second, in line with our "real-time" scenario, we had our model make a prediction for a five-minute interval, calculate the RMSE, and then updated the weights before moving on to the next timestep. The two methods produced similar RMSE values during model validation, but the former had to be trained on our dataset twenty times (to simulate a larger amount of data than we actually have) compared to just once for the "real-time" model.

We also tested whether or not splitting the models by AM 6-11 and PM 4-9 periods had any effect, and we observed that the "static" model performed just as well while the "online" model improved significantly.

We used PyTorch to implement our model as an alternative to Tensorflow. Both environments are similar (with Tensorflow being primarily developed by Google and PyTorch by Facebook), but the authors chose to use PyTorch because we found it to be more intuitive to build models with.

Our final predictions are charted in Figures 2 and 3.

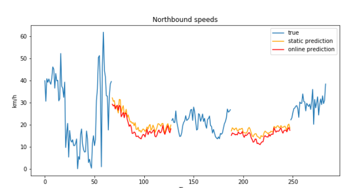

Figure 2: Northbound speed predictions.

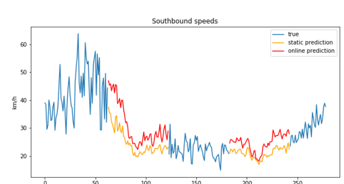

Figure 3: Southbound speed predictions.