7 Traffic prediction

7. (15 points) You were just hired by Ways, and they ask you to use machine-learning methods to predict the traffic at intersections in a city.

   (a) The first thing you have to decide is how to encode the inputs. The first city you analyze data for is designed on a grid, where all streets either run due north/south or due east/west. You know the names of all the streets in the city in advance. You have measurements of the traffic at many intersections at many times of day, and you have to encode each measurement using a vector of feature values.

   For each of the features below, explain how best to encode it. Your encoding should be able to take into account both the fact that traffic at different points on the same road tends to be correlated and the fact that some whole parts of a city might be busier than others.

   i. Name of street that runs north/south

   ii. Name of street that runs east/west

   > **Solution:** Use one-hot encoding for each street. These features are important because the street name can help generalization (there is lots of traffic on Mass Ave, but not on the alleyway next door).

   iii. Latitude (positive real)

   iv. Longitude (positive real)

   > **Solution:** Treat each as a positive real value. Technically longitude is a cyclic quantity, so you'd have trouble if your city were Greenwich, UK, but we did not expect you to handle that here.

   v. Day of week (an integer from 1 to 7)

   > **Solution:** One choice might be a binary feature for weekend or not. One-hot is also good. Integers are not good, because the implied ordering doesn't make sense (Monday is not "closer" to Sunday than to Wednesday.)

   vi. Time of day (a real number between 0 and 24)

   > **Solution:** Full credit for noting that there is a wrap-around problem even if not correctly handled. Ideal would be to represent it in 2D by mapping into range $0 - -2\pi$ and then use the sin and cos as features. Another reasonable answer is to discretize into bins.

   (b) You do such a good job in that city, you are assigned to work on a new city that is very old, and although there are still only two streets involved in an intersection, the streets wind all around and don't run in a consistent direction, so it's not clear which order to put them in. So, a training example for an intersection might be arbitrarily described as one of

```
("1st street", "park street", 34.4, 54.2, 02139, 22.35, Friday)
("park street", "1st street", 34.4, 54.2, 02139, 22.35, Friday)
```

You can assume that for any two streets, there is only a single intersection between them. How would you change your encoding of the first two features to deal with this, or would you leave it as is? Explain your answer.

> **Solution:** We create a bag-of-words-like encoding. Namely, for each of the n street in the city, we assign an index, and the intersection of streets indexed i and j is represented as an n-dimensional vector of zeros with ones at indices i and j.
>
> It is important that your representation still be able to efficiently represent "there is a lot of traffic on Mass Ave at 5PM on weekdays". If you do one-hot encoding of intersections, you can't generalize well.

(c) Your goal is to predict how much traffic there is at a given intersection at a given time, so now you have to think about how to encode the output. We'll just consider a single direction of a single road. The raw data that you have contains either a positive number or `None`. If it's a number, that is the average speed in miles per hour of the cars that went through this road in the past 10 minutes. If it is `None`, then no cars went through in the last 10 minutes.

Design an output encoding for this data. Specify how many dimensions it has, and precisely what transformation you would do on the raw output values in order to compute your encoding.

> **Solution:** When we wrote this question, we assumed that the only thing that would cause the case that "no cars went through in the last 10 minutes" was that there were no cars on the road at all (so the effective speed was infinite, or the speed limit. However, some students pointed out that this condition could also happen when the traffic was at a standstill, in which case the speed is 0.
>
> We gave up and gave everyone full credit.
>
> In the case that None means traffic is at a standstill, you can just convert None to 0 and encode as a numeric feature. If it means that there are no cars at all, then we need to do something special. One good solution is to use two features: one for whether this value is None, and one encoding the speed if not. Another good one is to convert None to the speed limit.

(d) You are using a neural network to predict traffic using the encoding you selected above. What activation function(s) would you use on your output units, and what loss function would you use? Explain your choices.

> **Solution:** Depends on choice above. If one-hot bins, then softmax/NLLM. If positive numeric values then relu/squared loss. If general (positive or negative values) then the identity function and squared loss.