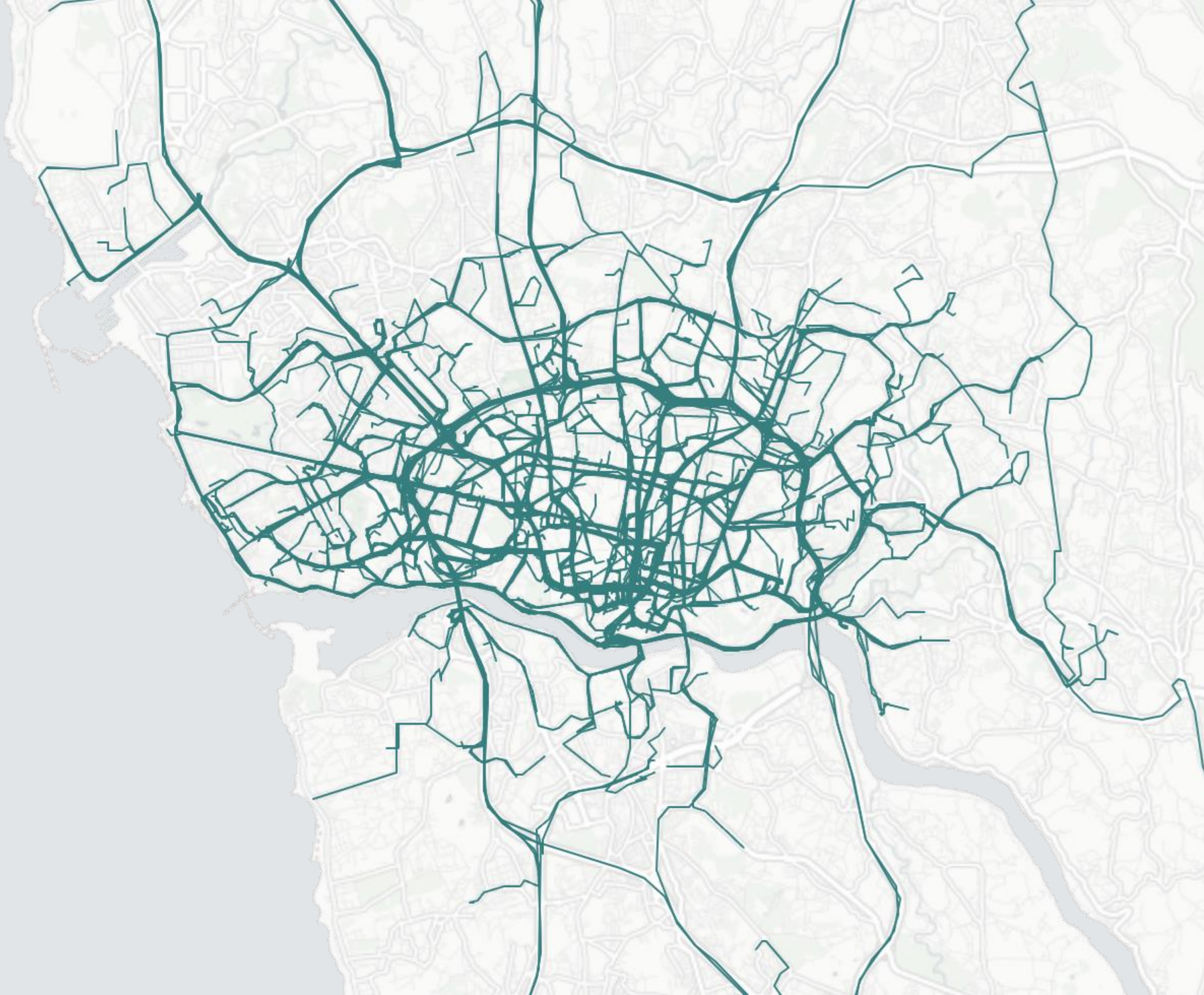
Trajectory Prediction & Similarity Search

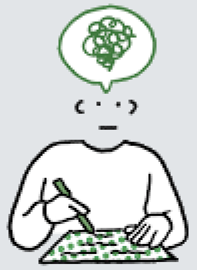
Jaz Zhou
jz878@drexel.edu

NY taxis -Stephen Wiltshire

Taxi Trajectory data



Use and Intention



Urban Design

Next POI prediction can reveal future traffic bottleneck, help with traffic flow optimization.

Similar trajectory search can form a base for trajectory clustering, which can help urban designer identify different pattern of traffic, which can help planning better location for public infrastructure, or the future city structure in general

Overview

- RNN (Recurrent Neural Network)
  - Learn and Predict next location
  - Generate lower-dimensional trajectory embeddings
- KD-Tree (K-Dimensional Tree)
  - Store and Index trajectory embeddings
  - Conduct similarity search

Raw Taxi Trajectory data

Porto, Portugal

Over 1,680,000 trajectories (use 20,000 in my project)

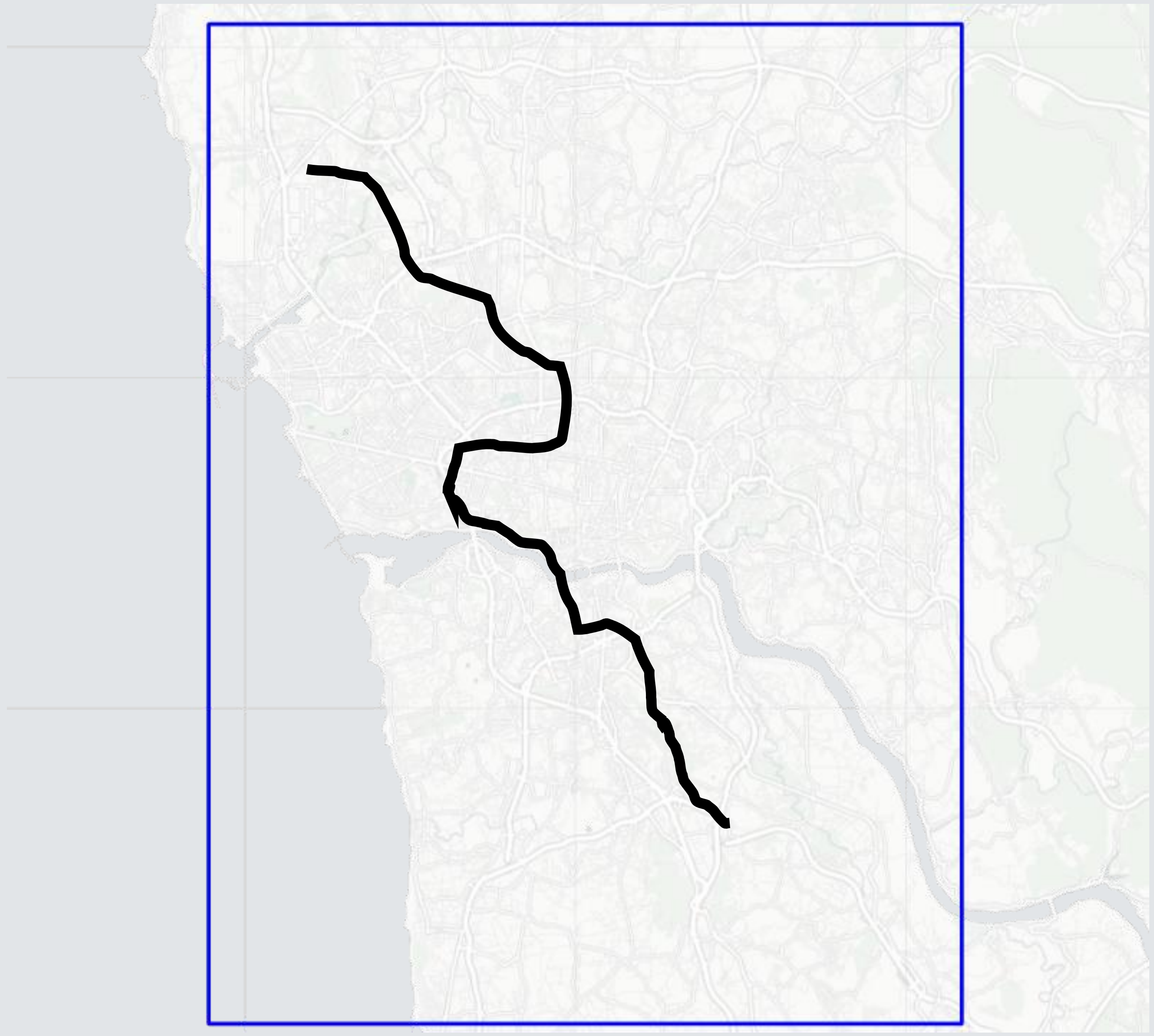
Available: *https://kaggle.com/competitions/pkdd-15-predict-taxi-service-trajectory-i*

One trajectory: list of coordinates in the form of latitude and longitude pairs.

```
[
        [-8.618643,41.141412],[-8.618499,41.141376],[-8.620326,41.14251],[-8.622153,41.143815],
        [-8.623953,41.144373],[-8.62668,41.144778],[-8.627373,41.144697],[-8.630226,41.14521],
        [-8.632746,41.14692],[-8.631738,41.148225],[-8.629938,41.150385],[-8.62911,41.151213],
        [-8.629128,41.15124],[-8.628786,41.152203],[-8.628687,41.152374],[-8.628759,41.152518],
        [-8.630838,41.15268],[-8.632323,41.153022],[-8.631144,41.154489],[-8.630829,41.154507],
        [-8.630829,41.154516],[-8.630829,41.154498],[-8.630838,41.154489]
]
```

One-hot Vector

How do we represent one trajectory as an input to RNN?

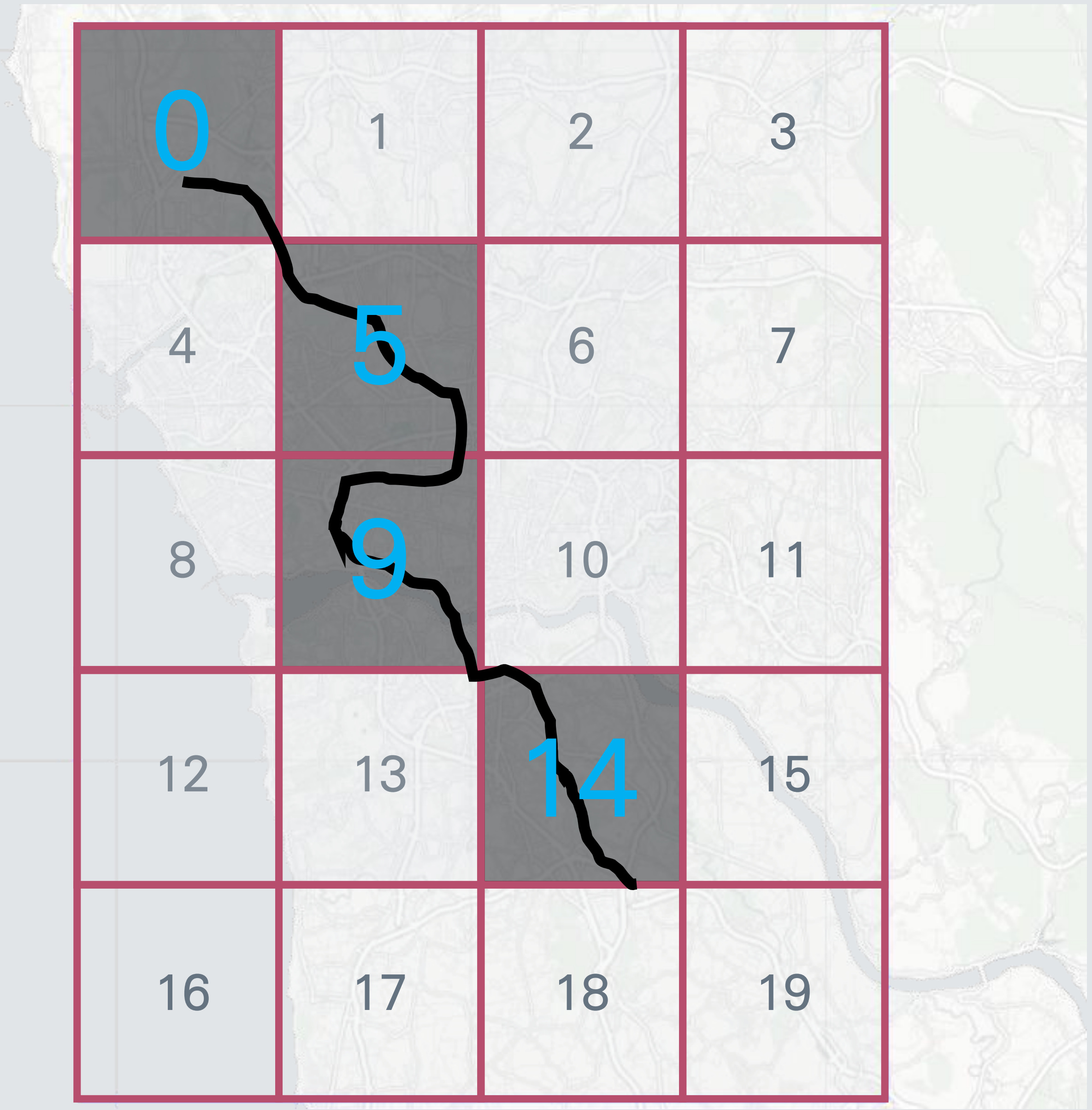


One-hot Vector

How do we represent one trajectory as an input to RNN?

Break map into cells
Each cell has an index

$$\begin{bmatrix} 0 \\ 5 \\ 9 \\ 14 \end{bmatrix}$$



223 rows x 168 cols, each cell 100m x 100m

One-hot Vector

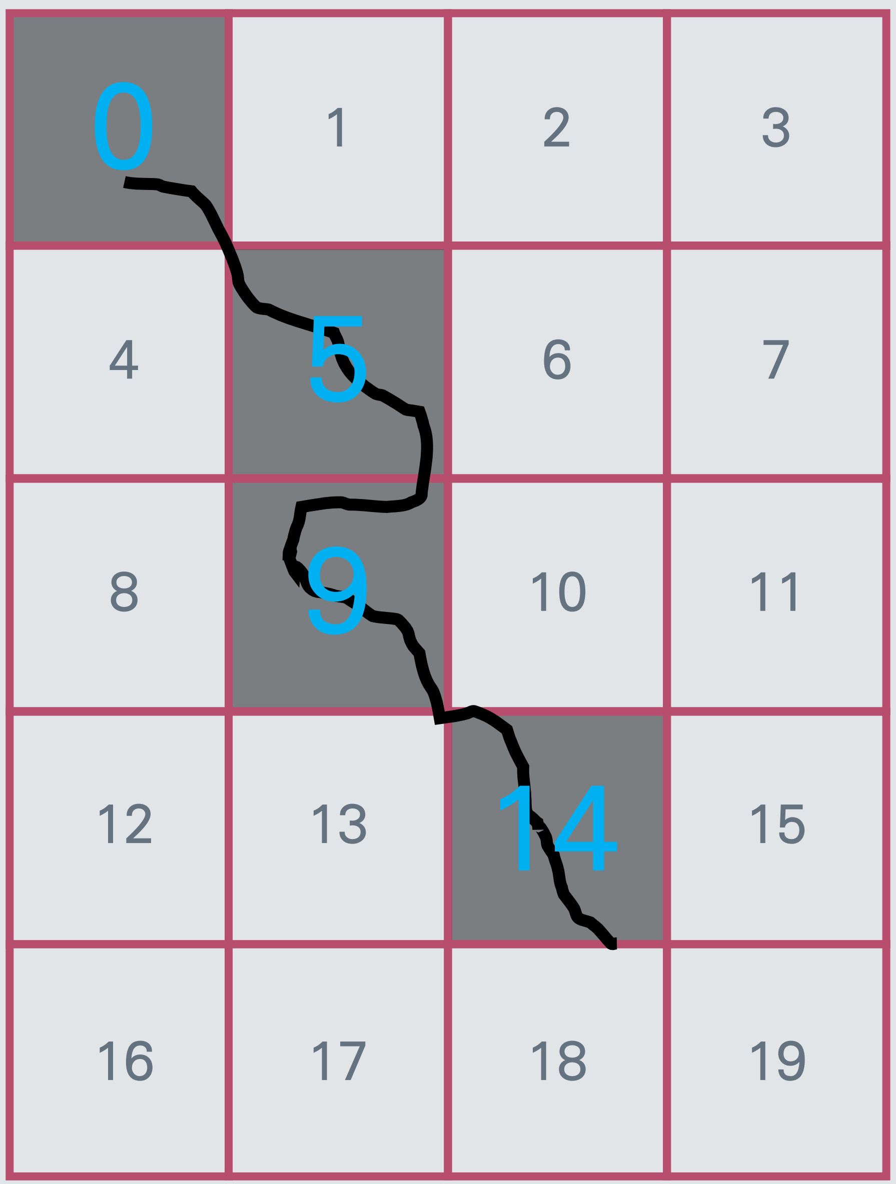
[1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]

[0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]

[0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]

[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0]

$$\begin{bmatrix} 0 \\ 5 \\ 9 \\ 14 \end{bmatrix}$$



(223*168)-dimensional vector

Next Cell Prediction

[1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]

↓ ?

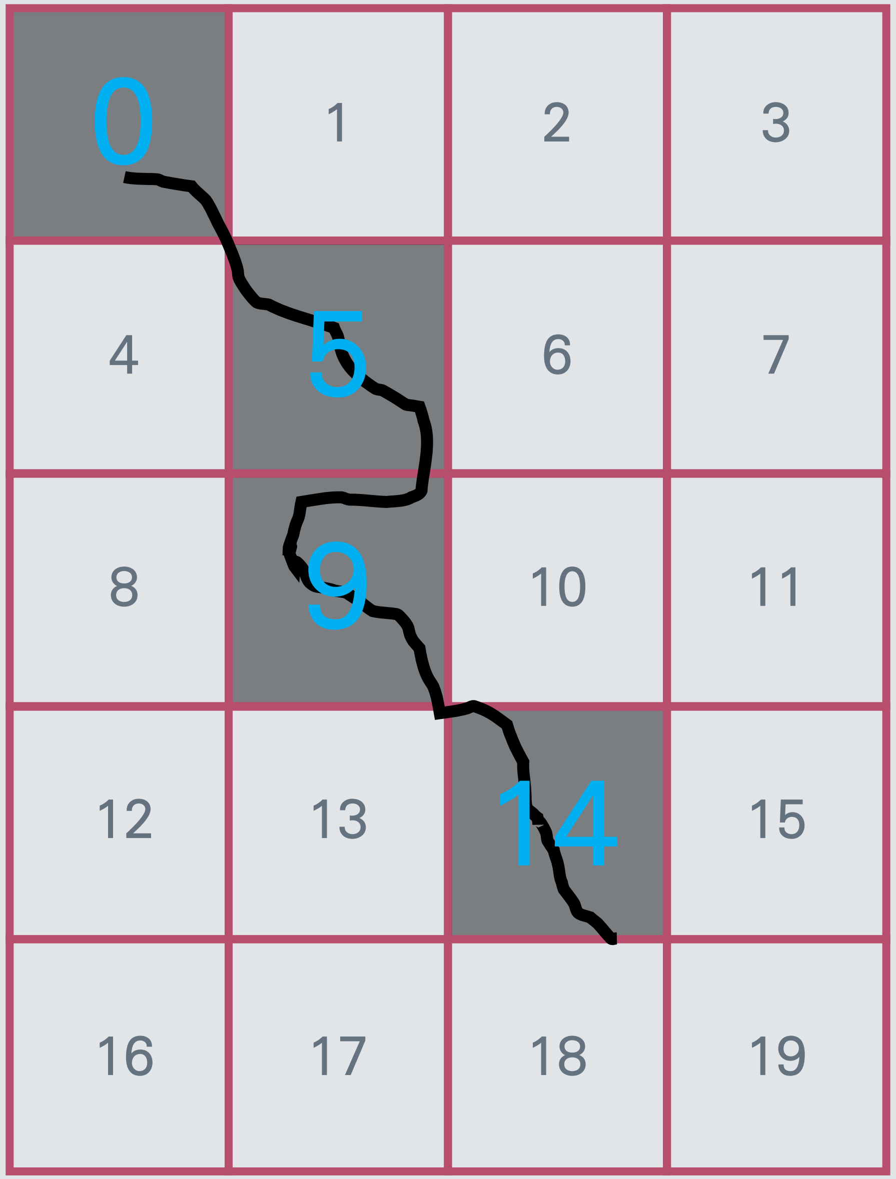
[0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]

↓ ?

[0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]

↓ ?

[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0]



Next Cell Prediction

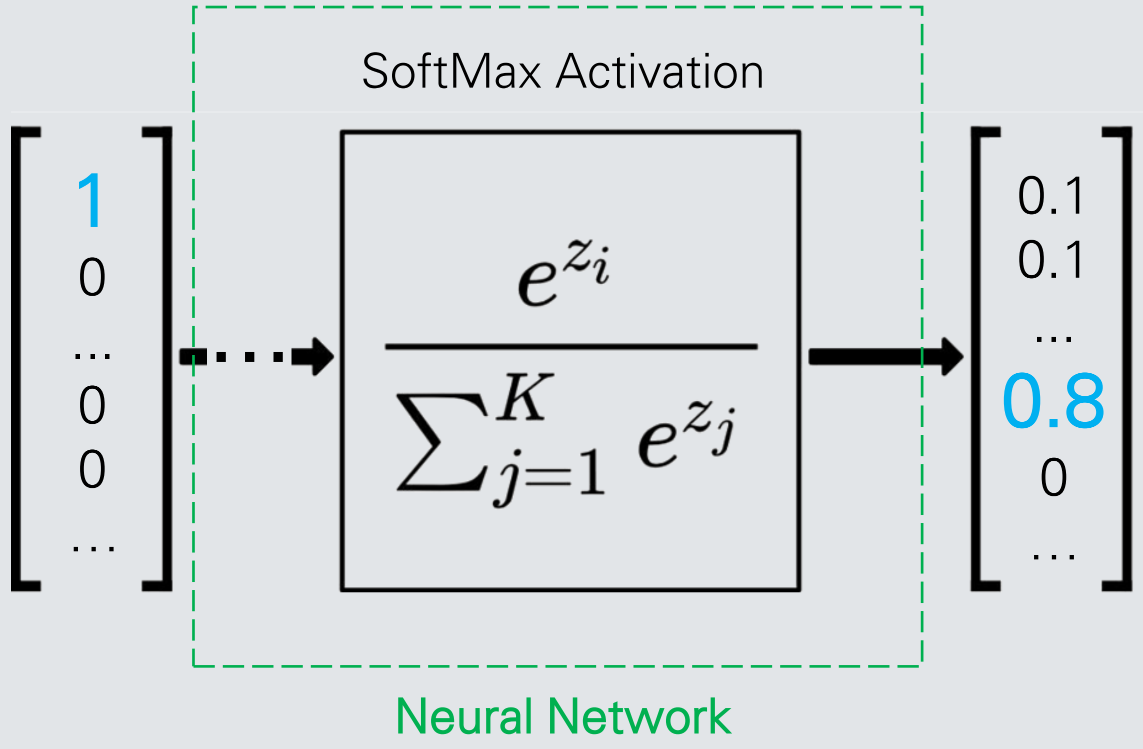
[1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]

[0.1, 0.1, 0, 0, 0, 0.8, 0, … 0.2, 0.01, 0.01, 0.01, 0]

[0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]

[0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]

[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0]



Next Cell Prediction

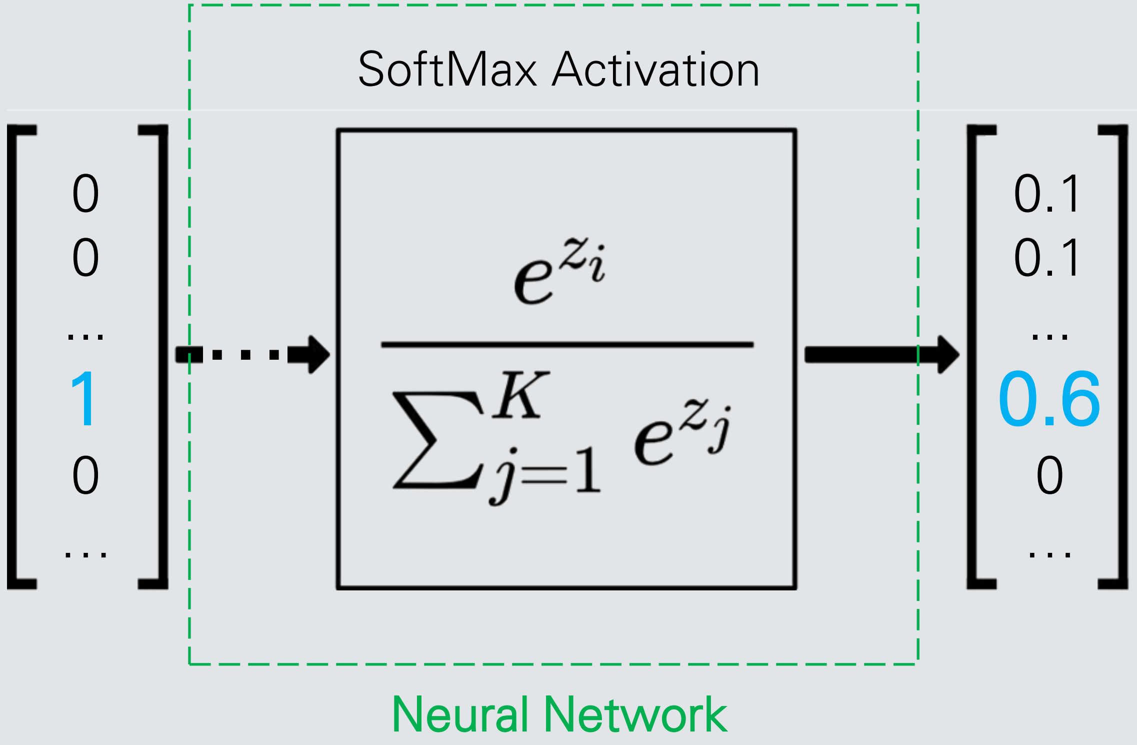
[1 , 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]

[0, 0, 0, 0, 0, 1 , 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]

[0.1, 0.1, 0, 0, 0, … 0, 0.6, 0.2, 0.01, 0.01, 0.01, 0]

[0, 0, 0, 0, 0, 0, 0, 0, 0, 1 , 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]

[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1 , 0, 0, 0, 0, 0]



Next Cell Prediction

[1 , 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]

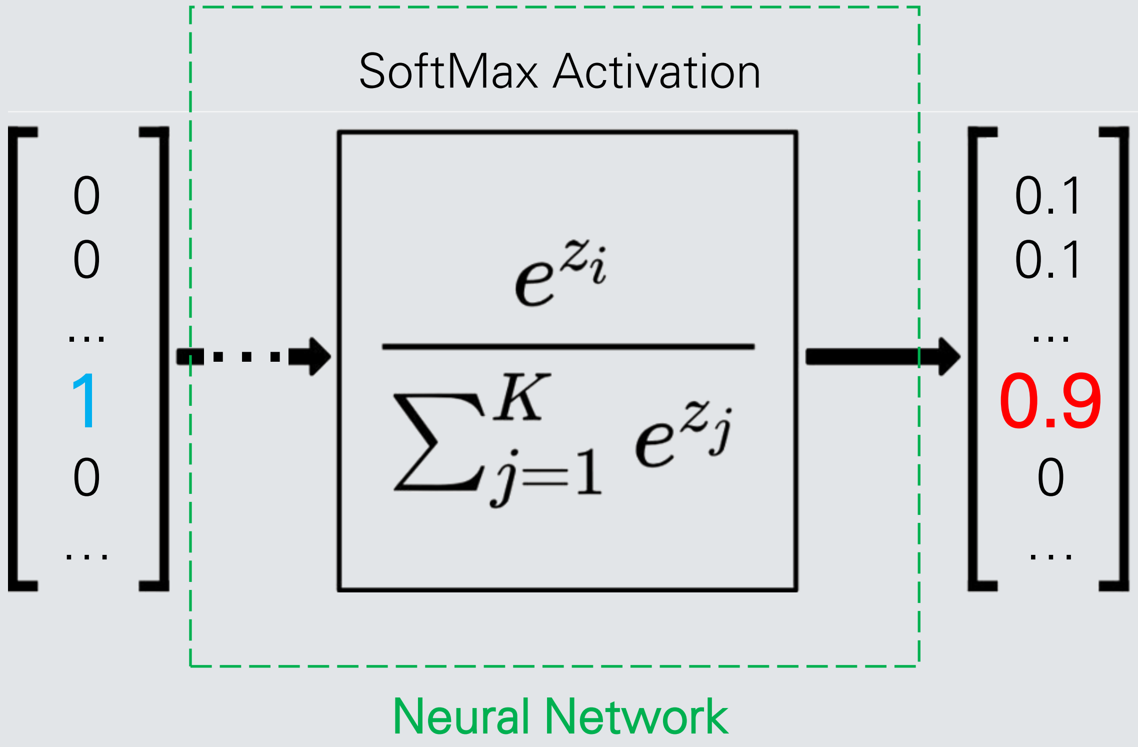
[0, 0, 0, 0, 0, 1 , 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]

[0, 0, 0, 0, 0, 0, 0, 0, 0, 1 , 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]

[0.1, 0.1, 0, 0, 0, 0.9, 0, … 0.2, 0.01, 0.01, 0.01, 0]

[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1 , 0, 0, 0, 0, 0]

Loss


SoftMax Activation
$$\begin{bmatrix} 0 \\ 0 \\ \cdots \\ 1 \\ 0 \\ \cdots \end{bmatrix} \dashrightarrow \boxed{\frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}}} \rightarrow \begin{bmatrix} 0.1 \\ 0.1 \\ \cdots \\ 0.9 \\ 0 \\ \cdots \end{bmatrix}$$
Neural Network

Next Cell Prediction

[1 , 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]

[0.1, 0.1, 0, 0, 0, 0.8, 0, … 0.2, 0.01, 0.01, 0.01, 0]

[0, 0, 0, 0, 0, 1 , 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]

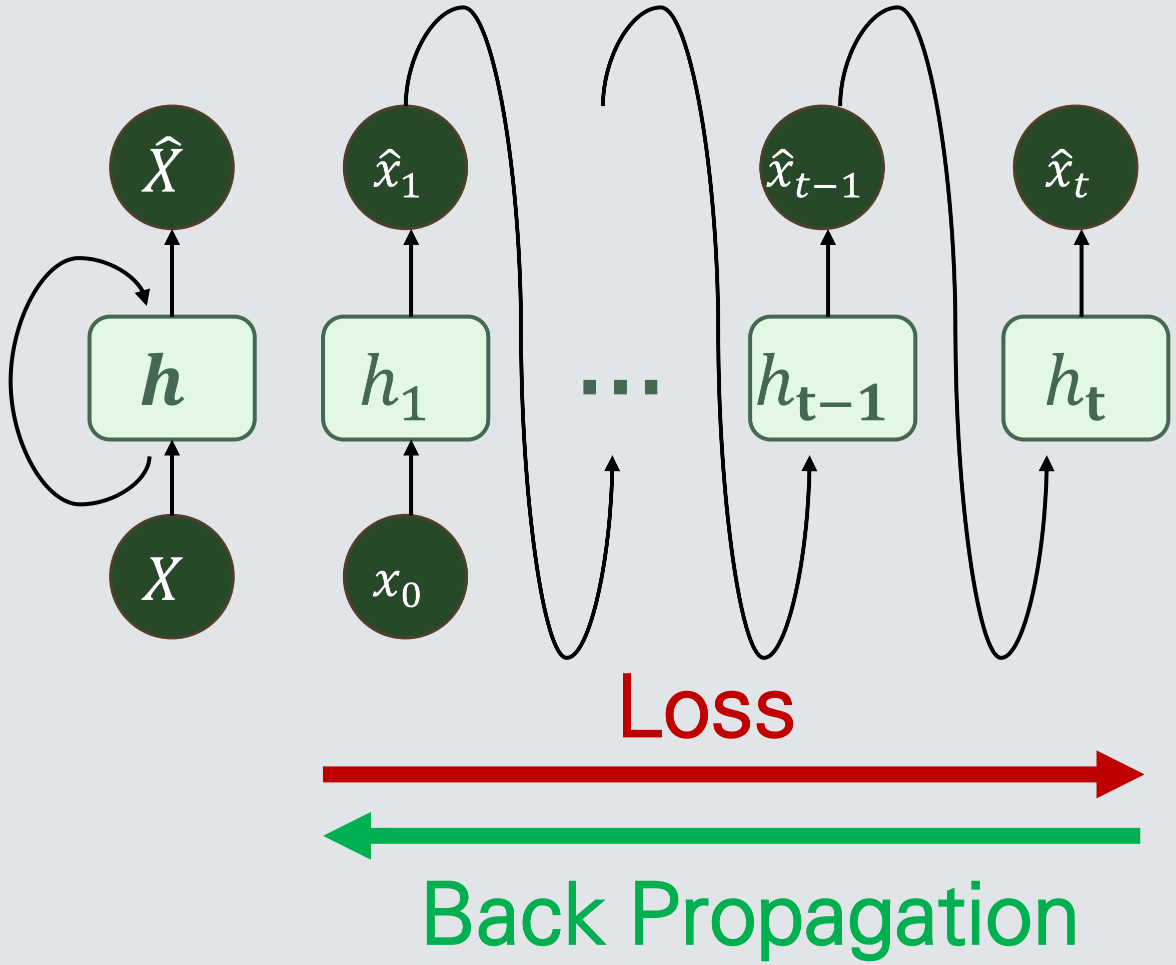
[0.1, 0.1, 0, 0, 0, … 0, 0.6, 0.2, 0.01, 0.01, 0.01, 0]

[0, 0, 0, 0, 0, 0, 0, 0, 0, 1 , 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]

[0.1, 0.1, 0, 0, 0, 0.8, 0, … 0.2, 0.01, 0.01, 0.01, 0]

[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1 , 0, 0, 0, 0, 0]

Loss

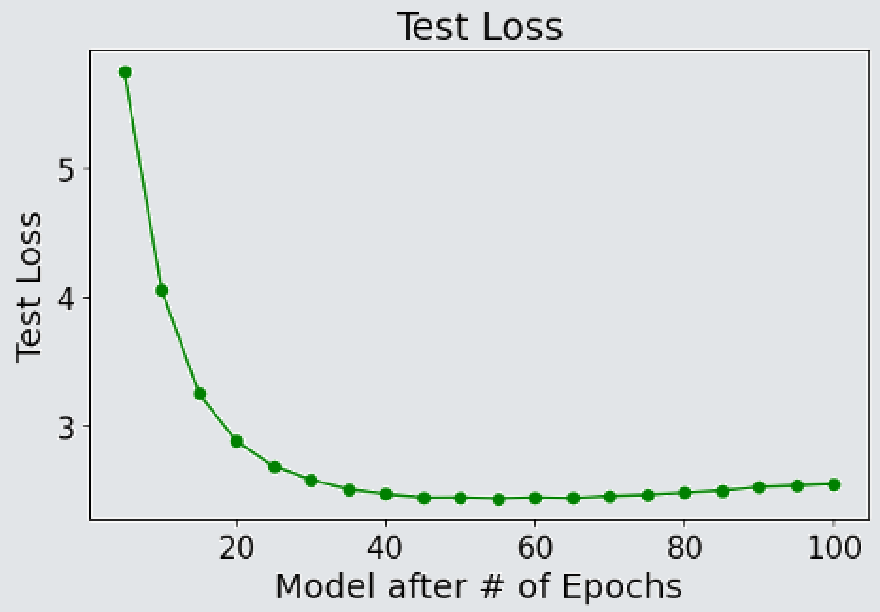


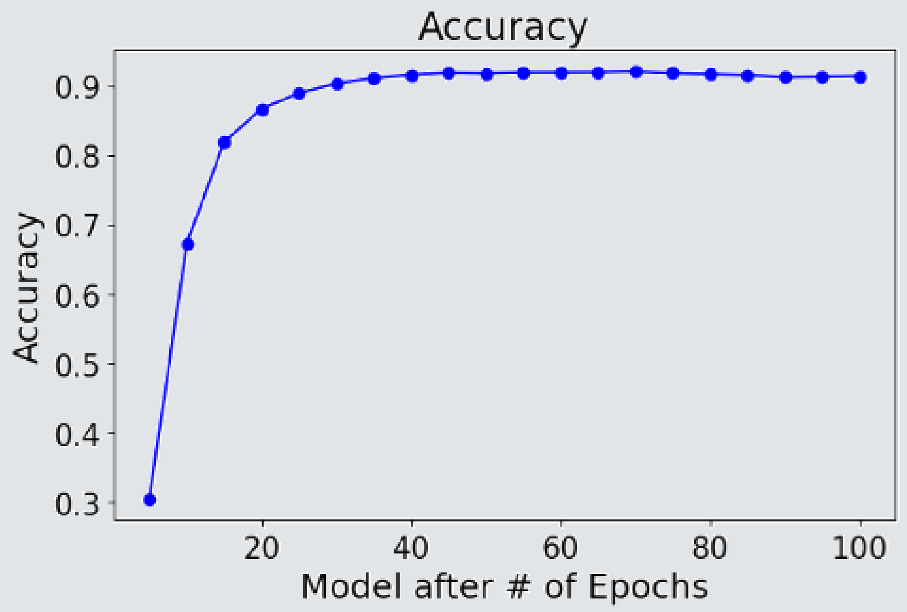
Next Cell Prediction

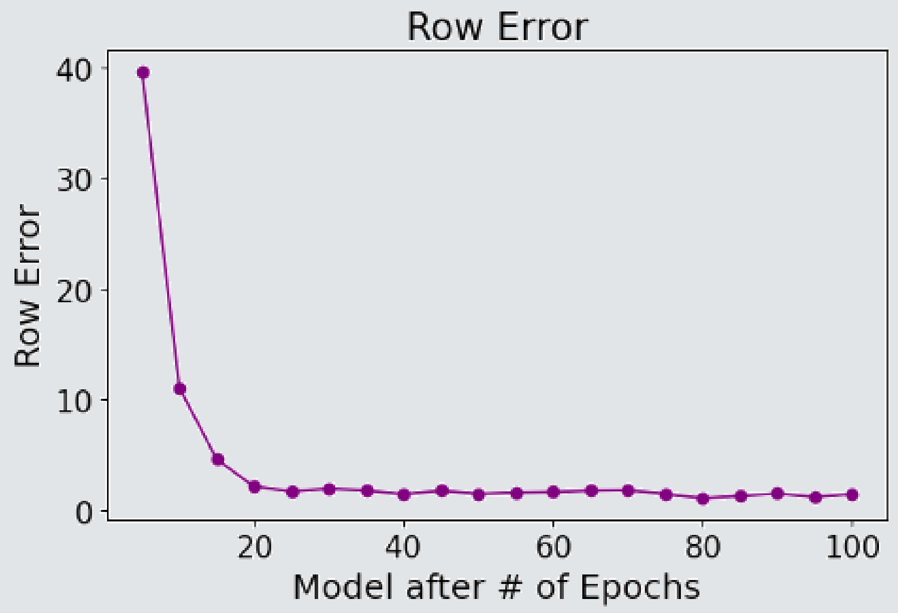
rnn construction
2 hidden layer
32 neuron/layer

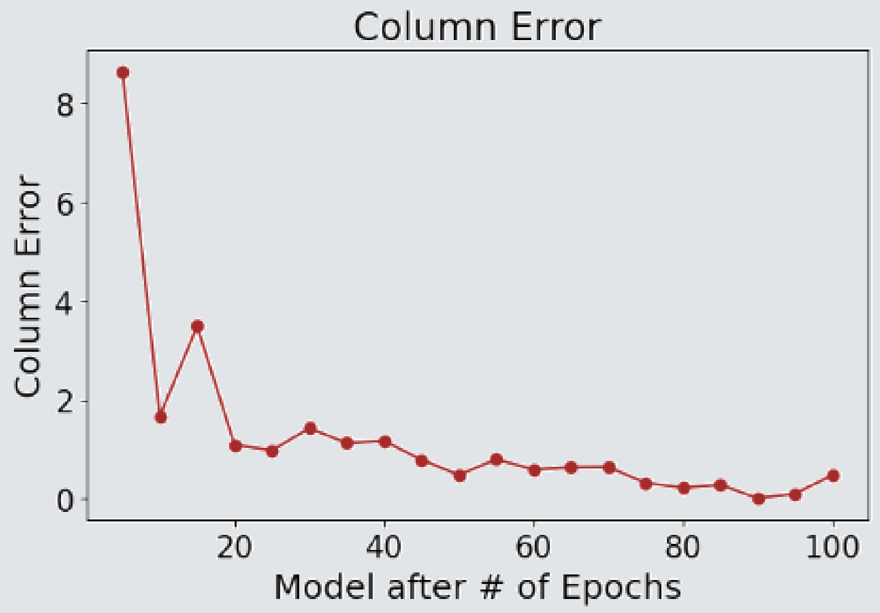
Train and validate(100 epoch)
Train: 8000 trajectories
Validate: 2000 trajectories

Test (20 models)
Test: 10000 trajectories









Similarity Search

[1, 0, 0, 0, 0, 0, 0, 0, 0, …, 0, 0, 0, 0, 0, 0, 0, 0, 0]

...

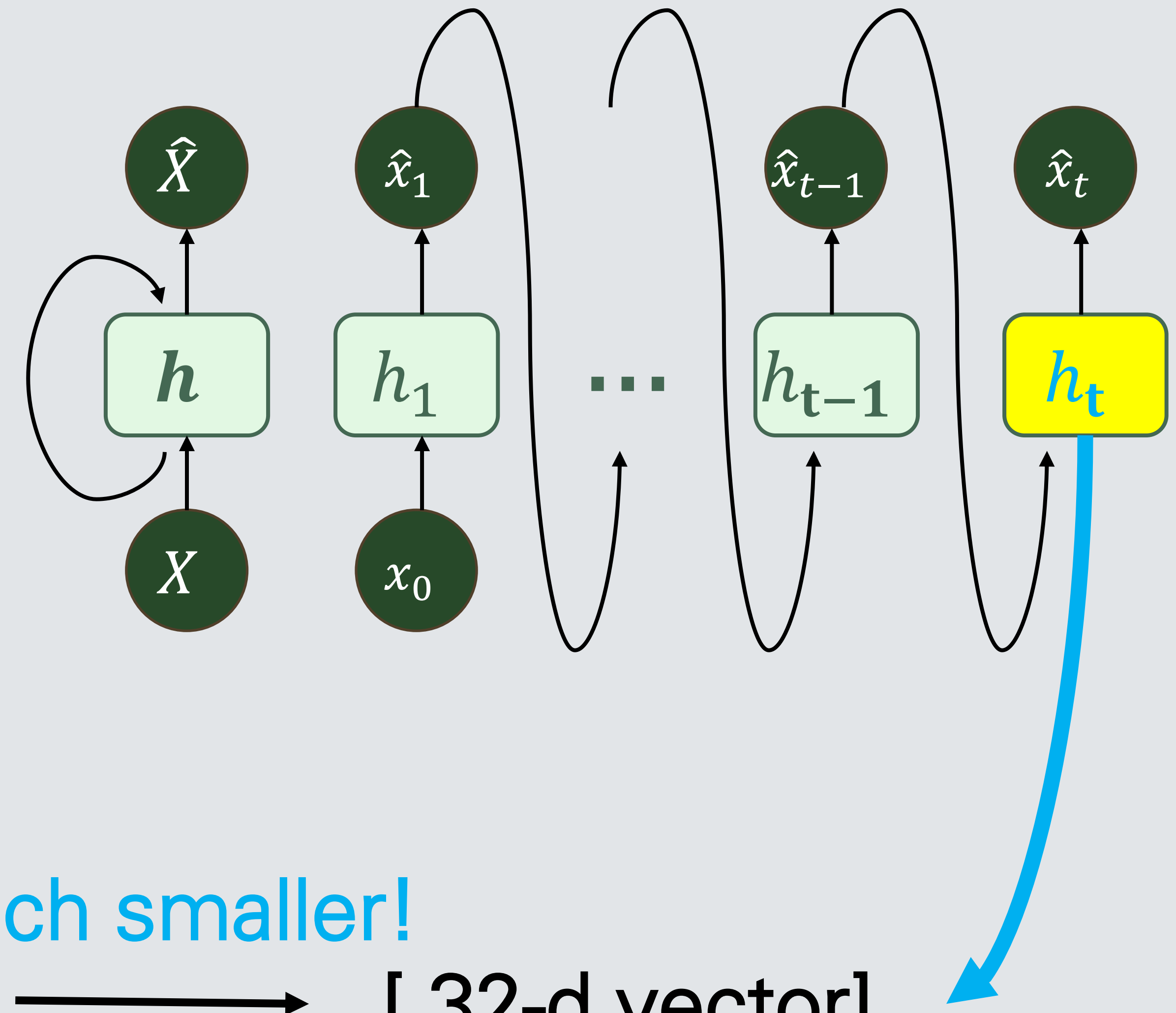
[0, 0, 0, 0, 0, 1, 0, 0, 0, …, 0, 0, 0, 0, 0, 0, 0, 0, 0]

...

[0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, …, 0, 0, 0, 0, 0]

...

[0, 0, 0, 0, 0, 0, 0, 0, 0, …, 0, 0, 0, 1, 0, 0, 0, 0, 0]

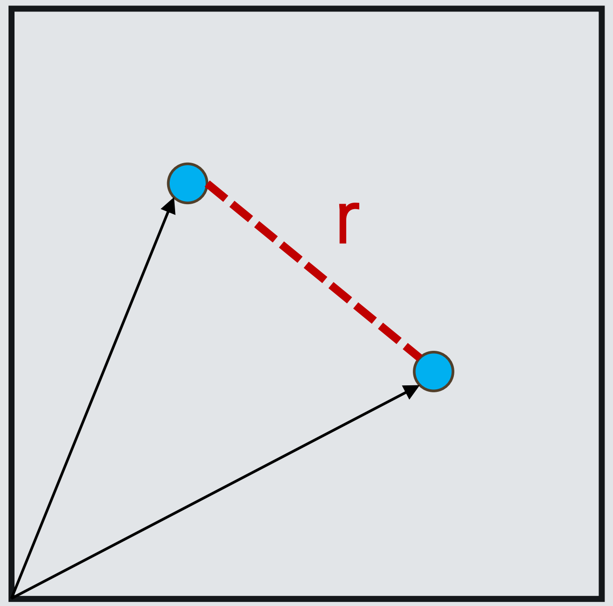


Much smaller!

[(223*168) X (#cells) matrix] $\longrightarrow$ [32-d vector]

Similarity Search

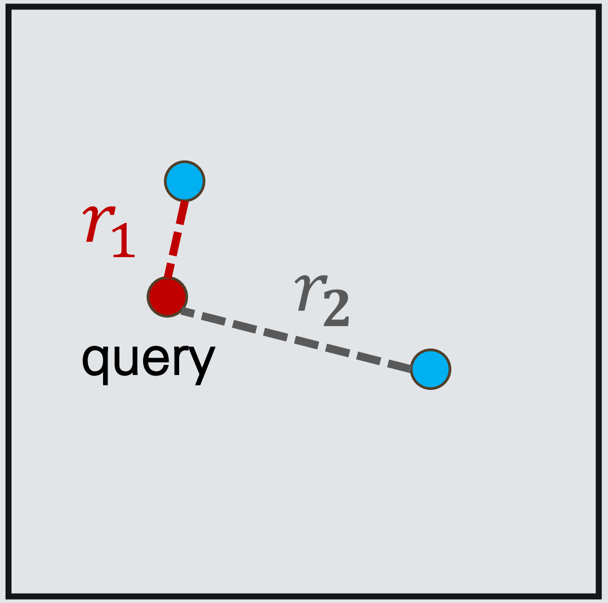
$$\theta(\boldsymbol{m})$$



32D

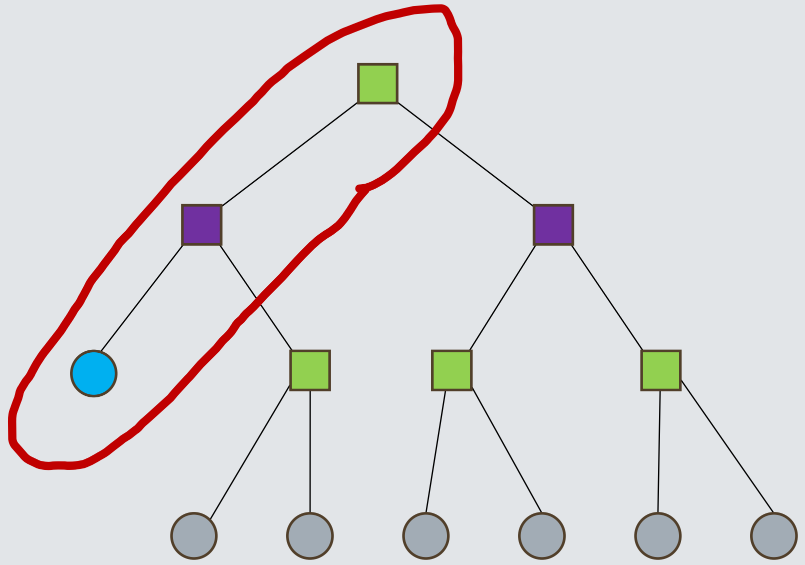
Similarity Search

$$\theta(\boldsymbol{mn})$$



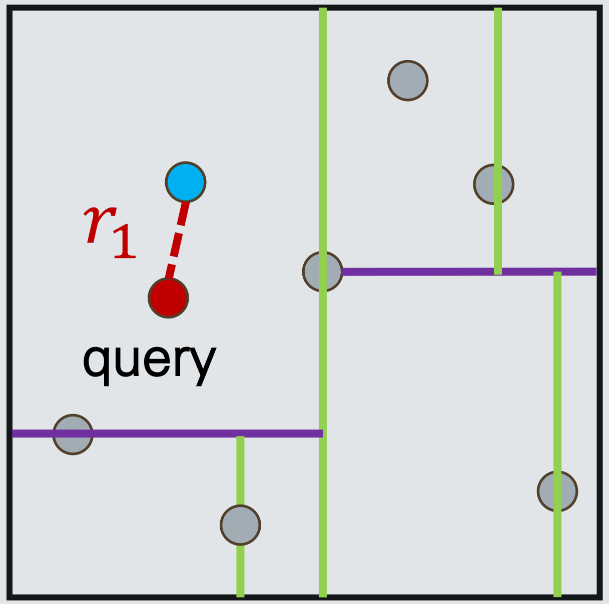
m = 32

Similarity Search



kd-tree

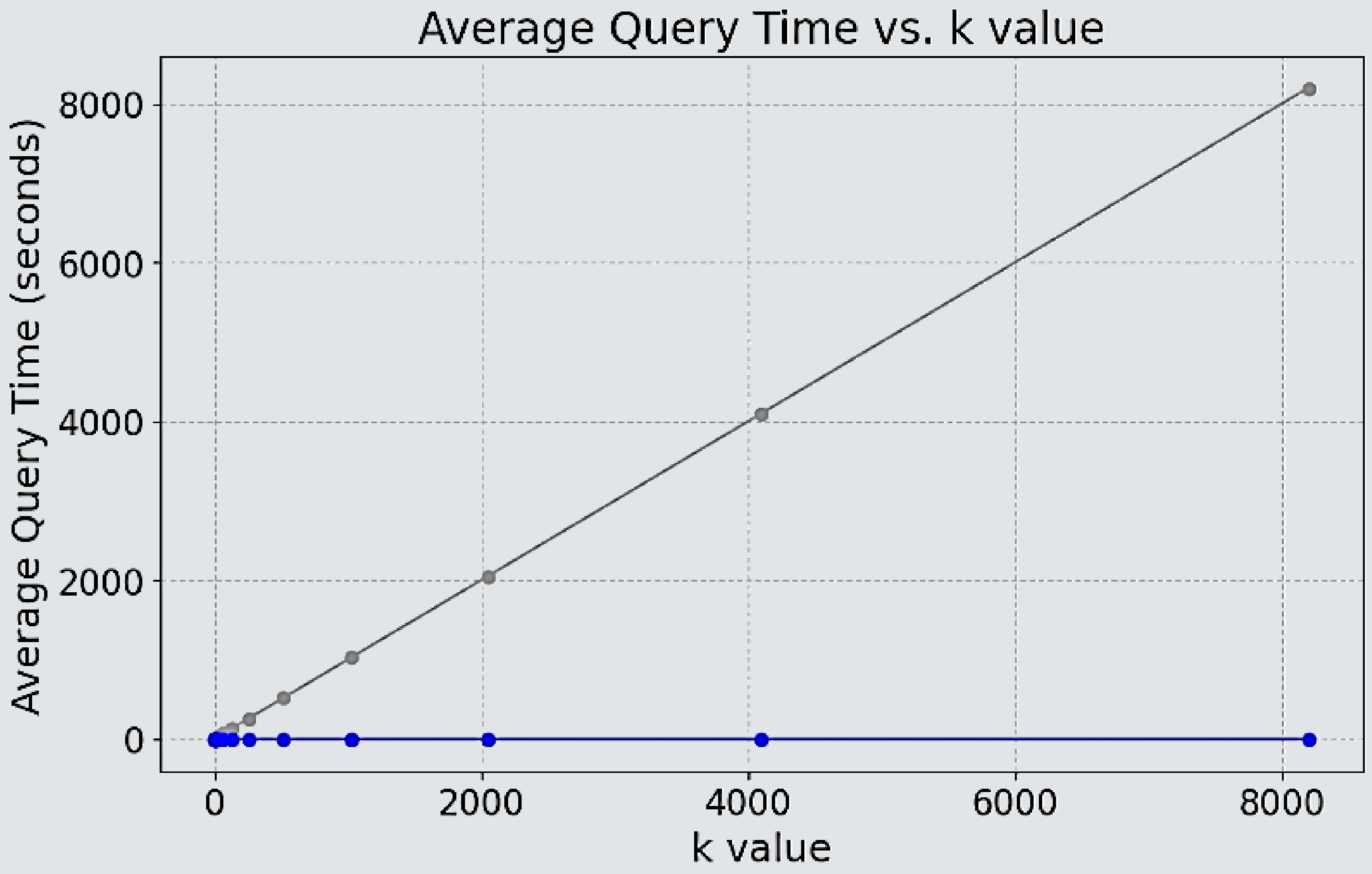
$$\theta(\boldsymbol{m} \lg n)$$



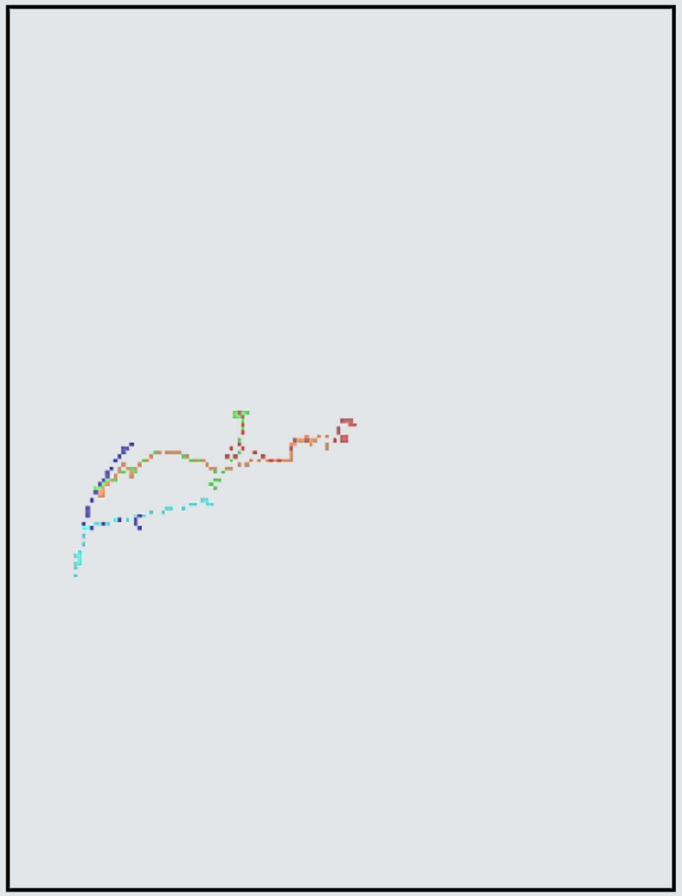
m = 32

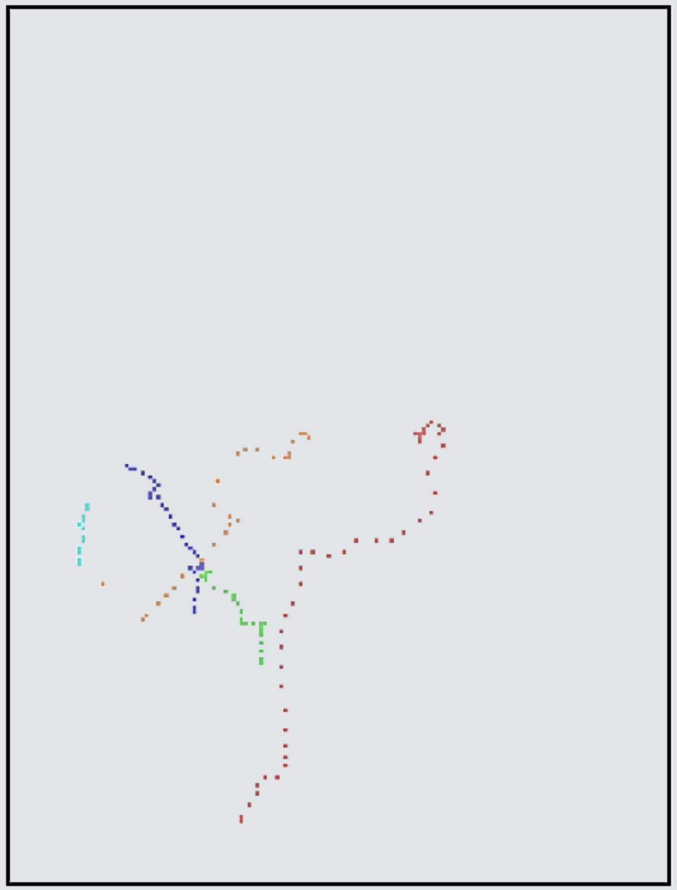
k-nearest neighbor query efficiency

```
k value Average Query Time (seconds)
1       0.000301671028137207
2       0.0002997875213623047
4       0.00019993782043457032
8       0.0003003120422363281
16      0.0002978801727294922
32      0.0002997398376464844
64      0.0002016782760620117
128     0.0002980947494506836
256     0.0004020214080810547
512     0.000499558448791504
1024    0.0006979465484619141
2048    0.0011017322540283203
4096    0.001997971534729004
8192    0.0029003143310546873
```



Similarity Search





Individual Skills and Contributions

- Architecture background
    - Identify key areas of focus in urban design
    - Recognize the application or research value of chosen topic

- I did everything!
    - EDA
    - Preprocessing data for ML
    - Use PyTorch to train a prediction model (e.g., RNN)
    - Use KD-Tree lib for indexing and efficient search

NY taxis -Stephen Wiltshire