AI for Traffic Prediction: Project Presentation Phase II

Quentin Fruytier

Reminder (of problem):

Description:

Predict future traffic speed/conditions by leveraging previously observed data.

Impact:

- ❖ Vehicle Routing Problem
- ❖ Vehicle Scheduling problem
- ❖ HeadLights Algorithms

$$[\boldsymbol{X}^{(t-T'+1)}, \cdots, \boldsymbol{X}^{(t)}; \mathcal{G}] \xrightarrow{h(\cdot)} [\boldsymbol{X}^{(t+1)}, \cdots, \boldsymbol{X}^{(t+T)}]$$

$\mathcal{G}$: Road sensor network

$\boldsymbol{X}^{(t)}$: Road traffic conditions at time t

Li, Y., Yu, R., Shahabi, C., & Liu, Y. (2018, February). *DIFFUSION CONVOLUTIONAL RECURRENT NEURAL NETWORK: DATA-DRIVEN TRAFFIC FORECASTING.* ICLR 2018. https://arxiv.org/pdf/1707.01926.pdf

Reminder (of goals):

Previous Goals:

- ❖ **Understand**, **reproduce** and **improve** state of the art non-parametric methods for traffic forecasting.

Previous proposed Methodology

- ❖ Make effective use of external factors as well as combining ideas from successful methods.

Progress since:

- ❖ Investigated DCRNN more thoroughly.
- ❖ I like the intuitive approach
- ❖ I believe in its problem representation

DCRNN: Diffusion Convolution Neural Network

- ❖ Deep learning framework for traffic forecasting
- ❖ Incorporates both spatial and temporal dependency in the traffic flow
  - ➢ Spatial dependency using bidirectional random walks on the graph and convolution
  - ➢ Temporal dependency using RNN in encoder-decoder architecture and scheduled sampling.

Li, Y., Yu, R., Shahabi, C., & Liu, Y. (2018, February). *DIFFUSION CONVOLUTIONAL RECURRENT NEURAL NETWORK: DATA-DRIVEN TRAFFIC FORECASTING.* ICLR 2018. https://arxiv.org/pdf/1707.01926.pdf

Data:

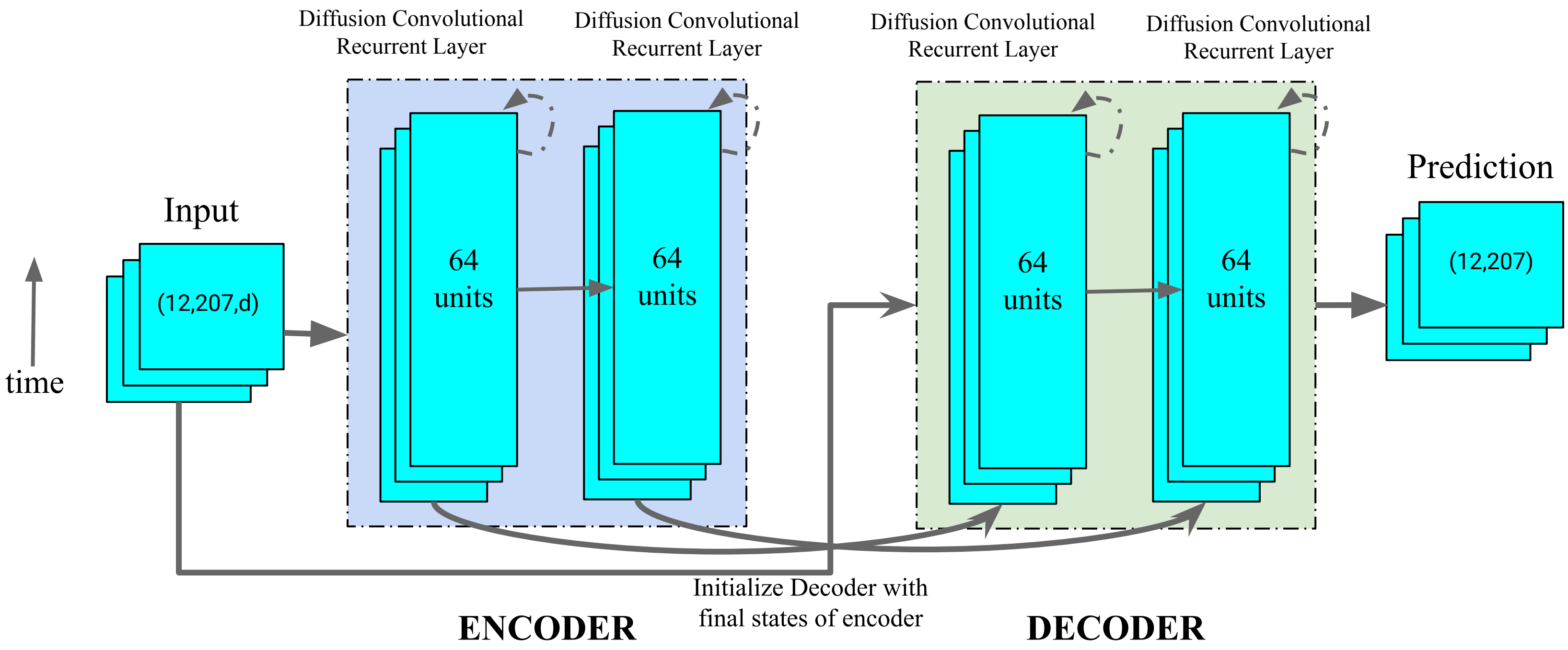
METR-LA Data Set:

- ~30 000 time series data points (5 min intervals)
- 207 road sensors
- traffic speeds at sensors
- time of the week
- time of day
- Sensor Network Data

Pre-processing:

- sensor network adjacency matrix
- sequence length of 12
- normalize traffic speeds
- data shape: (~30 000, 12, 207, 3)
- train size: 23974
- validation size: 3425
- test size: 6850

Implemented Architecture



Diffusion Convolution

$$\boldsymbol{X}_{:,p} \star_{\mathcal{G}} f_{\boldsymbol{\theta}} = \sum_{k=0}^{K-1} \left(\theta_{k,1} \left(\boldsymbol{D}_O^{-1} \boldsymbol{W} \right)^k + \theta_{k,2} \left(\boldsymbol{D}_I^{-1} \boldsymbol{W}^\intercal \right)^k \right) \boldsymbol{X}_{:,p} \quad \text{for } p \in \{1, \cdots, P\}$$

Where $\boldsymbol{\theta} \in \mathbb{R}^{K \times 2}$ are the parameters of the filter.

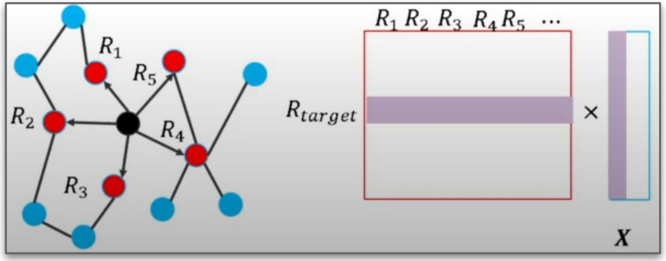
$\boldsymbol{X} \in \mathbb{R}^{N \times P}$ is the input, where N is the number of nodes (sensors) and P is the feature dimension

W is the downstream adjacency matrix where Wij is a function of the road distance between the two sensors

W^T is the upstream

Source: Li, Y., Yu, R., Shahabi, C., & Liu, Y. (2018, February). *DIFFUSION CONVOLUTIONAL RECURRENT NEURAL NETWORK: DATA-DRIVEN TRAFFIC FORECASTING*. ICLR 2018. https://arxiv.org/pdf/1707.01926.pdf

$$\boldsymbol{D}_O^{-1} \boldsymbol{W} \boldsymbol{X}_{:,p}$$



Source:【Review01】Diffusion convolutional recurrent neural Network: Data-driven Traffic Forecasting [Video file]. (2019, August 28). Retrieved April 07, 2021, from https://www.youtube.com/watch?v=lN8HrGFK3l0&t=709s

Implementation

Tensorflow 2.4.1 | Google Colab | Hardware Accelerator - GPU

```
class Encoder(tf.keras.Model):
      def __init__(self,_):
      def initialize_initial_state (self):
      def call(self, x, hidden):
```

```
class DCGRUCell(tf.keras.layers.AbstractRNNCell):
tf.compat.v1.nn.rnn_cell.MultiRNNCell()
tf.keras.layers.RNN()
```

```
class Decoder(tf.keras.Model):
      def __init__(self,_):
      def initialize_initial_state (self,
encoder_state):
      def call(self, x, hidden):
```

```
class DCGRUCell(tf.keras.layers.AbstractRNNCell):
tf.compat.v1.nn.rnn_cell.MultiRNNCell()
tf.keras.layers.StackedRNNCells()
tfa.seq2seq.BasicDecoder()
```

```
tf.keras.optimizers.Adam
with tf.GradientTape() as tape:
gradients = tape.gradient()
optimizer.apply_gradients()
```

Inspired from: https://www.tensorflow.org/addons/tutorials/networks_seq2seq_nmt

Initialization + training

Model weights:

```
tf.keras.initializers.GlorotNormal()
tf.constant_initializer()
Initialized as 0 otherwise
375296 parameters to train
```

Learning rate schedule

```
tf.keras.optimizers.schedules.PiecewiseConstantDecay([2000, 30000], [0.01, 0.001, 0.0001])
```

Results:

Evaluation:
Mean Absolute Error:

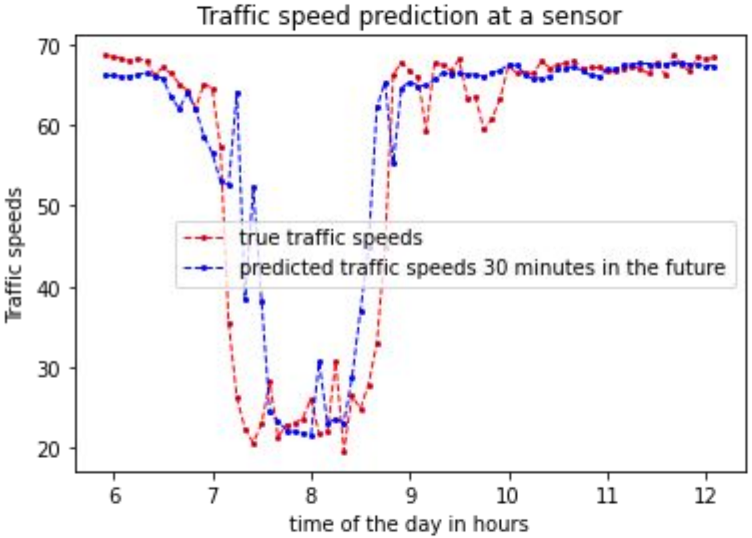
$$\mathrm{MAE} = \underbrace{\frac{1}{n}\sum_{i=1}^{n}}_{\text{test set}} |\underbrace{y_i}_{\text{predicted vaue}} - \underbrace{\hat{y}_i}_{\text{actual value}}|$$

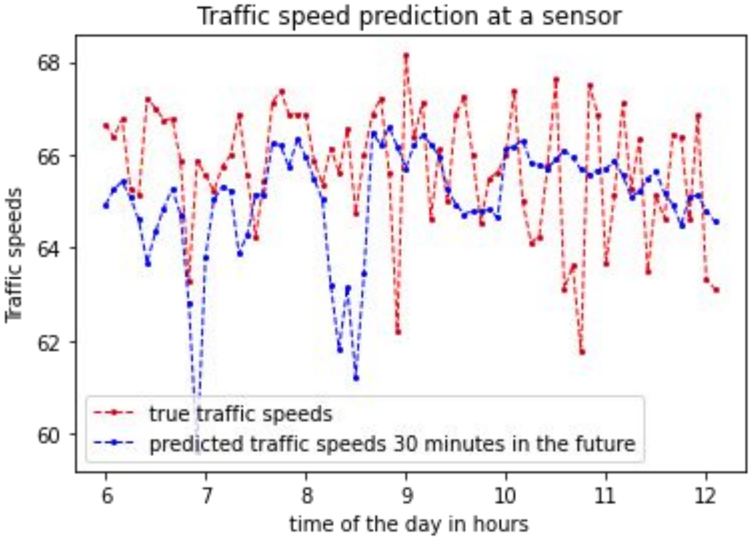
*100 epochs of training

Time Horizons	Metric	FC-LSTM [1]	My Implementation		
			Just Traffic Speeds	Traffic Speeds + Time of Day	Traffic Speeds + Time of Day + Time of Week
15 mins	**MAE**	3.44	2.92	2.94	3.05
30 mins	**MAE**	3.77	3.41	3.39	3.48
60 mins	**MAE**	4.37	4.12	3.98	4.02

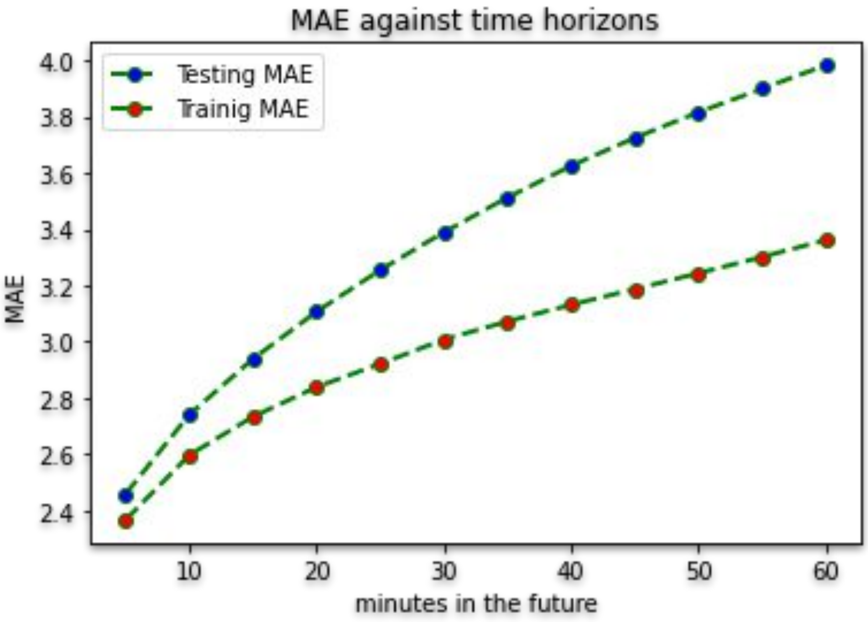
[1] Li, Y., Yu, R., Shahabi, C., & Liu, Y. (2018, February). *DIFFUSION CONVOLUTIONAL RECURRENT NEURAL NETWORK: DATA-DRIVEN TRAFFIC FORECASTING.* ICLR 2018. https://arxiv.org/pdf/1707.01926.pdf

Results (after only 20 epochs)

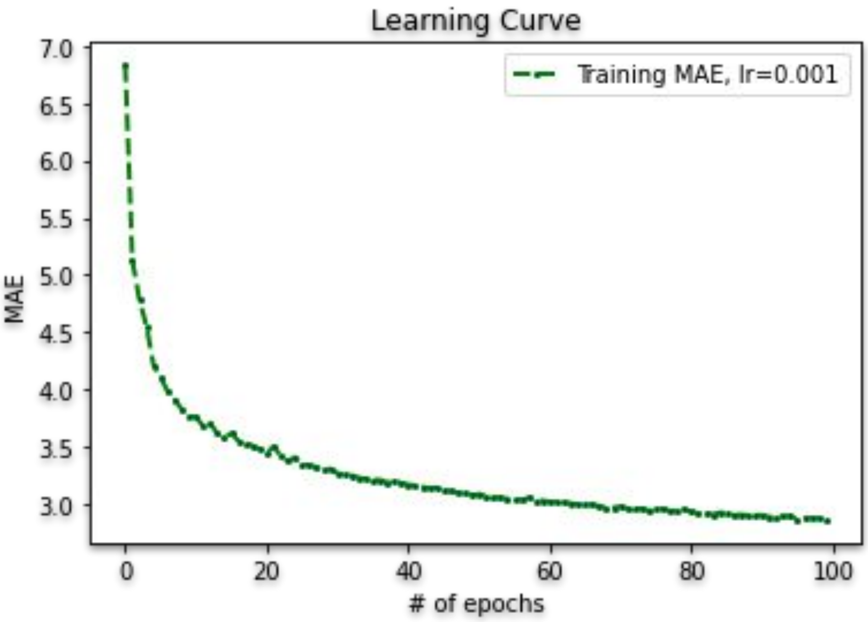




Results:



MAE against time horizons
Testing MAE
Trainig MAE
MAE
minutes in the future
4.0
3.8
3.6
3.4
3.2
3.0
2.8
2.6
2.4
10
20
30
40
50
60



Learning Curve
Training MAE, lr=0.001
MAE
# of epochs
7.0
6.5
6.0
5.5
5.0
4.5
4.0
3.5
3.0
0
20
40
60
80
100

Future:

Improve:

- ❖ Add Attention Mechanism[1]
- ❖ More in-depth evaluation (smoothness, how well model deals with peak hours, RMSE, etc.)
- ❖ Experiment with external factors
  - ➢ QTraffic Data set has Road Queries Data which have been shown to correlate with traffic speeds

[1] https://ojs.aaai.org//index.php/AAAI/article/view/3881

QUESTIONS?!