Long-Term Mobile Traffic Forecasting Using Deep Spatio-Temporal Neural Networks

Бузанов Никита
Moscow State University
8 мая 2023 г.

Оглавление

Введение

Введение

Наша цель – сделать точные долгосрочные прогнозы объёма трафика мобильной передачи данных, потребляемого пользователями в различных точках города, на основе предварительных измерений. Мы формально выражаем потребление мобильного трафика в масштабах всей сети, наблюдаемое за интервалом времени T, как пространственно-временную последовательность данных $D = \{D_1, D_2, ..., D_T\}$, где D_t - моментальный снимок объёма мобильного трафика в момент времени t по городу:

$$D_t = \begin{bmatrix} d_t^{(1,1)} & ... & d_t^{(1,Y)} \\ ... & ... & ... \\ d_t^{(X,1)} & ... & d_t^{(X,Y)} \end{bmatrix} \tag{1}$$

$d_t^{(x,y)}$ - объём трафика данных в квадратной ячейке с координатами (x, y).

Таким образом, нашу последовательность данных можно рассматривать как тензор $D \in \mathbb{R}^{T \times X \times Y}$. С точки зрения машинного обучения, задача пространственно-временного прогнозирования трафика заключается в прогнозировании наиболее вероятной K-ступенчатой последовательности точек данных, учитывающих S предыдущих наблюдений. Это означает, что нам нужно найти

$$D'_{t+1}, ..., D'_{t+K} = \underset{D_{t+1},...,D_{t+K}}{\operatorname{argmax}} \; p(D_{t+1}, ..., D_{t+K} | D_{t-S+1}, ..., D_t) \tag{2}$$

Заметим, что $d_{t+1}^{(x,y)}$ в значительной степени зависит только от трафика в соседних клетках, поэтому информацией связанной с удалёнными ячейками можно пренебречь. Следовательно, при прогнозировании трафика в $d_{t+1}^{(x,y)}$ ограничимся рассмотрением трафика на квадрате $(r+1) \times (r+1)$ с центром в точке (x, y).

Следовательно:

$$p(D_{t+1}|D_{t-S+1},...,D_t) \approx \prod_{x=1}^{X}\prod_{y=1}^{Y} p\left(d_{t+1}^{(x,y)}|F_{t-S+1}^{(x,y)},...,F_t^{(x,y)}\right) \tag{3}$$

где

$$F_t^{(x,y)} = \begin{bmatrix} d_t^{(x-\frac{r}{2},y-\frac{r}{2})} & ... & d_t^{(x+\frac{r}{2},y-\frac{r}{2})} \\ ... & d_t^{(x,y)} & ... \\ d_t^{(x-\frac{r}{2},y+\frac{r}{2})} & ... & d_t^{(x+\frac{r}{2},y+\frac{r}{2})} \end{bmatrix} \tag{4}$$

представляет собой матрицу трафика данных в момент времени t в области $(r+1)\times(r+1)$, с центром в точке (x,y). Тогда предсказание D_{t+1} может быть выражено как множество

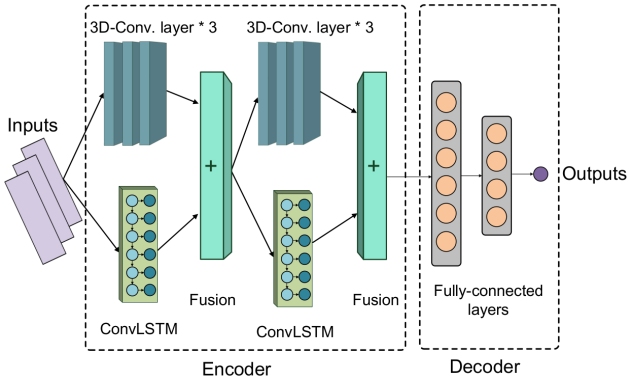
$$\tilde{D}_{t+1} = \left\{\tilde{d}_{t+1}^{(x,y)}|x=1,...,X;y=1,...,Y\right\} \tag{5}$$

где $\tilde{d}_{t+1}^{(x,y)}$ это предсказание для $d_{t+1}^{(x,y)}$, которое можно найти как

$$\tilde{d}_{t+1}^{(x,y)} = \underset{d_{t+1}^{(x,y)}}{\operatorname{argmax}}\, p\left(d_{t+1}^{(x,y)}|F_{t-S+1}^{(x,y)},...,F_t^{(x,y)}\right) \tag{6}$$

STN

STN – пространственно-временная нейронная сеть, разработанная для анализа трафика. STN следует парадигме кодирования-декодирования, в которой мы объединяем две нейронные сети ConvLSTM и 3D-ConvNet. Эти нейронные сети снабжаются матрицами трафика, формально выраженными как в (4). Выходы этих сетей идут на вход декодера, который представляет собой многослойный персептрон (MLP). Декодер делает уже окончательные прогнозы как показано в (6).



ConvLSTM

LSTM - это специальная рекуррентная нейронная сеть, которая устраняет проблему затухающего градиента, характерную для RNN, путём введения набора "вентилей". Несмотря на это количество параметров всё равно остаётся большим. ConvLSTM решает эту проблему вводом операций свёртки.

$$
\begin{gathered}
i_t = \sigma(W_{xi} * X_t + W_{hi} * H_{t-1} + W_{ci} \cdot C_{t-1} + b_i) \\
f_t = \sigma(W_{xf} * X_t + W_{hf} * H_{t-1} + W_{cf} \cdot C_{t-1} + b_f) \\
C_t = f_t \cdot C_{t-1} + i_t \cdot tanh(W_{xc} * H_{t-1} + b_c) \\
o_t = \sigma(W_{xo} * X_t + W_{ho} * H_{t-1} + W_{co} \cdot C_t + b_o) \\
H_t = o_t \cdot tanh(C_t)
\end{gathered}
\tag{7}
$$

где $*$ - оператор свёртки, $\cdot$ - произведение Адамара. $W_{(..)}$ и $b_{(.)}$ - веса и смещения, полученные в результате обучения. $X_t, C_t, H_t, i_t, f_t, o_t$ - всё это трёхмерные тензоры.

3D-ConvNets

Трёхмерные свёртки позволяют модели STN улавливать межвременные корреляции. Именно поэтому в STN используется 3D-ConvNets. Дана последовательность пространственно-временных данных с N картами $X = \{X_1, ..., X_N\}$, выходные данные трёхмерного свёрточного слоя будут состоять из $H_1, ..., H_M$ свёрнутых карт:

$$H_m = act\left(\sum_{n=1}^{N} X_n * W_{mn} + b_m\right) \tag{8}$$

где $*$ - оператор трёхмерной свёртки, $act()$ - функция активации. В STN входные данные (измерения) сначала обрабатываются параллельно одним ConvLSTM и одной 3D-ConvNets, затем их выходные данные объединяются с помощью слоя слияния, который выполняет следующее поэлементное сложение:

$$H(\Theta_H; X) = h_C(\Theta_1; X) + h_L(\Theta 2; X) \tag{9}$$

где h_C и h_L - выходные данные 3D-ConvNets и ConvLSTM, а $\Theta_H = \{\Theta_1, \Theta_2\}$ обозначает набор их параметров (веса и смещения).

D-STN

D-STN

Как видно из рис.1 STN плохо справляется с долгосрочным прогнозированием. В случае многоступенчатых предсказаний можно использовать последние прогнозы в качестве входных данных для следующего шага прогнозирования. Введём новую нейронную сеть Double STN или D-STN, которая включает в себя два ключевых усовершенствования:

- Алгоритм Уробороса
- смесь STN с исторической статистикой

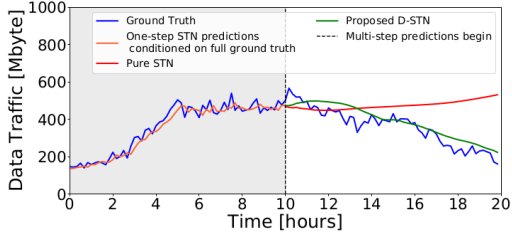


Рис.: 1

Алгоритм Уробороса

Algorithm 1 The Ouroboros Training Scheme

Inputs:
Time series training data $\mathcal{D} = \{D_1, D_2, ..., D_T\}$
Initialize:
A pre-trained model $\mathcal{M}$ with parameters Θ.
for $e = 1$ to E **do**
  for $t = 1$ to $T - S$ **do**
    if $t = 1$ **then**
      $Q \leftarrow \{D_1, D_2, ..., D_S\}$ ▶ Generate input queue using first S ground truth measurements.
    else
      Pop the first element out of Q,
      Predict D'_{S+t-1} by $\mathcal{M}$ with input Q,
      Push D'_{S+t-1} to the end of Q.
    end if
    Generate the target input $T \leftarrow D_{S+t}$.
    Train $\mathcal{M}$ with input Q and target T by SGD.
  end for
end for

Объединение прогнозов с исторической статистикой

Как видно из рис.2 трафик данных характеризуется определённой периодичностью как в ежедневном, так и в недельном циклах. А также, видно, что выборка близка к эмпирическому среднему. Таким образом, включение предварительных знаний о средних значениях в модель может улучшить эффективность прогнозирования.

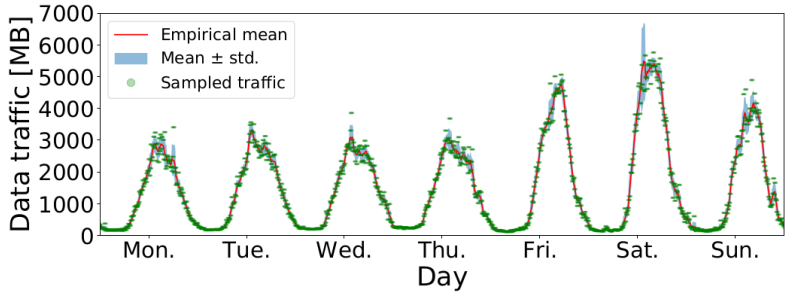


Рис.: 2

Предположим, что мы запускаем прогнозирование в момент времени t, обозначим за D^M_{t+h} и D^O_{t+h} предсказания в момент времени $t+h$ оригинального STN и STN с OTS соответственно. $\overline{D}_{t+h}$ – среднее значение фактических измерений. Тогда наше предсказание D-STN в h-ый час имеет вид:

$$D'_{t+h} = \gamma(h)\left[\alpha(h)D^M_{t+h} + (1-\alpha(h))D^O_{t+h})\right] + (1-\gamma(h))\overline{D}_{t+h} \tag{10}$$

где

$$\gamma(h) = 1 - \frac{1}{1+e^{-(wh+b)}} \quad \alpha(h) = max\left(1 - \frac{h(1-\delta)}{S}, \delta\right) \tag{11}$$

Заметим, что $\gamma(h) \to 0$ при $h \to \infty$, так как мы хотим чтобы эмпирическое среднее значение доминировало при больших h. На основе кросс-валидации по обучающим данным получено: $w = 0.01$, $b = -5$. Кроме того, на практике $\overline{D}_{t+h}$ может обновляться в режиме реального времени. Также замети, что чистый STN заслуживает большего веса на начальных этапах прогнозирования ($\delta = 0.5$ чтобы гарантировать, что D^M_{t+h} и D^O_{t+h} вносят равный вклад с течением времени).

Сравнение STN и D-STN с другими методами прогнозирования временных рядов

Сравнение STN и D-STN с другими методами прогнозирования временных рядов

Оценим эффективность STN и D-STN с широко используемыми методами прогнозирования временных рядов, а именно HW-ExpS с $\alpha = 0.9,\ \beta = 0.1,\ \gamma = 0.001$, ARIMA с $p = 3,\ d = 1,\ q = 2$, MLP с двумя слоями, ConvLSTM и 3D-ConvNets. Количественно оцениваем точность данных методов с помощью нормального среднеквадратичного значения:

$$NRMSE = \frac{1}{\overline{d}}\sqrt{\sum_{k=1}^{N}\frac{(\tilde{d_k} - d_k)^2}{N}} \tag{12}$$

где $\overline{d}$ - среднее значение d_i-ых.

На рис.3 видно, что STN и D-STN обеспечивают наилучшую точность. Тогда как HW-ExpS в значительной степени завышает/недооценивает будущий трафик, а ARIMA даёт почти линейные, медленно изменяющиеся оценки. NRMSE для D-STN меньше на 60% и 30% чем у HW-ExpS и ARIMA соответственно.

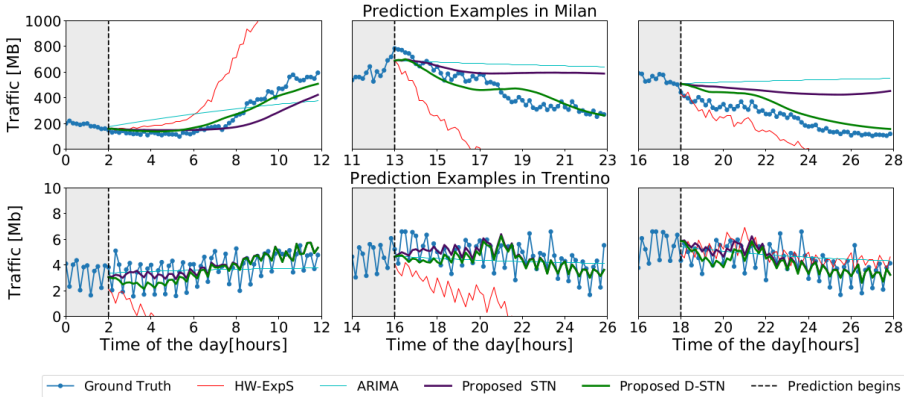


Рис.: 3

Кроме того, на рис.4 видно, что STN очень хорошо прогнозирует центр города, при этом немного недооценивает интенсивность на окраинах. HW-ExpS значительно переоценивает объём трафика.

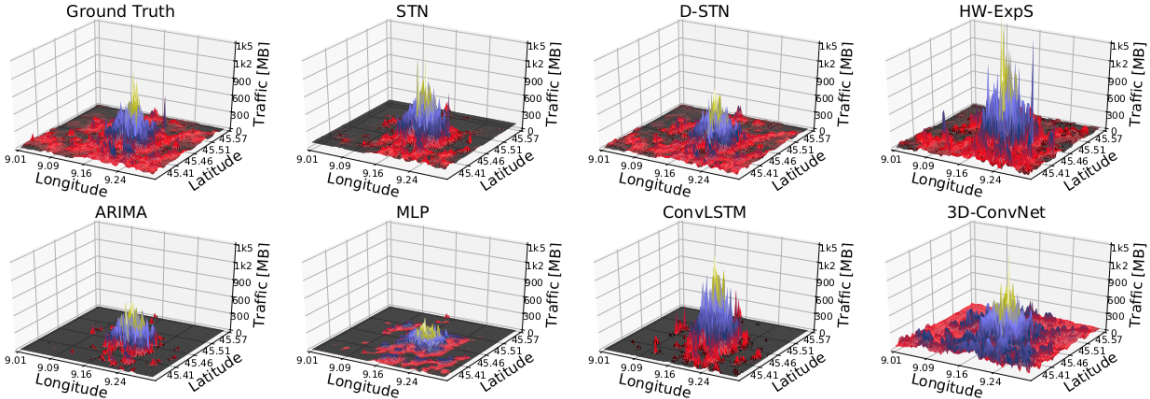


Рис.: 4

References

References

- Long-Term Mobile Traffic Forecasting Using Deep Spatio-Temporal Neural Networks (Оригинальная статья)
- Dataset
- GitHub Chaoyun Zhang и Paul Patras