Постепенное улучшение производительности Graph WaveNet на

|  |  |  |
| --- | --- | --- |
|  | Прогнозирование трафика |  |
| Сэм Шлейфер  Стэнфордский университет shleifer@stanford.edu | Клара МакКрири  Стэнфордский университет mccreery@stanford.edu  17 декабря, 2019  АБСТРАКТНЫЙ | Вамси Читерс  Стэнфордский университет vamsikc@stanford.edu |

Мы представляем ряд модификаций, которые улучшают ранее достигнутую производительность Graph WaveNet в задаче прогнозирования трафика METR-LA. Цель этой задачи — спрогнозировать будущую скорость трафика на каждом датчике в сети на основе показаний датчика за прошлый час. Граф WaveNet (GWN) — это пространственно-временная графовая нейронная сеть, которая чередует свертку графа для агрегирования информации от близлежащих датчиков и расширенные извилины для агрегирования информации из прошлого. Мы улучшаем GWN за счет (1) использования улучшенных гиперпараметров, (2) добавления связей, которые позволяют большим градиентам возвращаться к ранним сверточным слоям, и (3) предварительного обучения для более простой задачи краткосрочного прогнозирования трафика. Эти модификации уменьшают среднюю абсолютную ошибку на 0,06 в задаче METR-LA, что почти равно улучшению GWN по сравнению с его предшественником. Эти усовершенствования распространяются на набор данных PEMS-BAY с аналогичной относительной величиной. Мы также показываем, что создание отдельных моделей для краткосрочных и долгосрочных прогнозов еще больше повышает производительность. Код доступен на [https://github.com/sshleifer/ Graph-WaveNet.](https://github.com/sshleifer/Graph-WaveNet)

arXiv:1912.07390v1 [eess. SP] 11 дек 2019

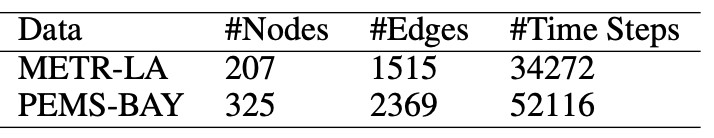
# ЗНАКОМСТВО

Американцы проводят за рулем в среднем 51 минуту в день и 11 минут в пробке.1 Улучшенное прогнозирование трафика может помочь свести к минимуму заторы на дорогах, предупреждая путешественников о задержках, чтобы они могли скорректировать свои маршруты или время отправления. Кроме того, идеи, полученные в результате улучшенного пространственно-временного моделирования, могут быть распространены на другие важные приложения, такие как экология, эпидемиология и климатология.

В этой статье основное внимание уделяется прогнозированию трафика: прогнозированию будущих скоростей трафика на каждом датчике в сети с учетом последних скоростей трафика на каждом датчике и пространственной информации. Дорожная сеть представлена в виде матрицы смежности, содержащей расстояние между датчиками на дороге.

# НАБОР ДАННЫХ

Мы используем тот же набор данных о трафике, что  *и Graph WaveNet* и его предшественник, *DCRNN,* о котором мы подробнее расскажем в разделе «Связанные работы»:



METR-LA: Этот набор данных поступает от датчиков на автомагистралях округа Лос-Анджелес, которые регистрируют скорость проезжающих транспортных средств. Показания каждого датчика разбиваются на 5-минутные блоки и впоследствии усредняются. Мы провели все эксперименты на наборе данных METR-LA, а затем убедились, что наши модификации повышают точность на более крупном, но аналогично структурированном наборе данных PEMS-BAY, который содержит 325 датчиков и данные за 6 месяцев из области залива.

1

2019 Городской Мобильность Сообщать [https://static.tti.tamu.edu/tti.tamu.edu/documents/ mobility-report-2019.pdf](https://static.tti.tamu.edu/tti.tamu.edu/documents/mobility-report-2019.pdf)

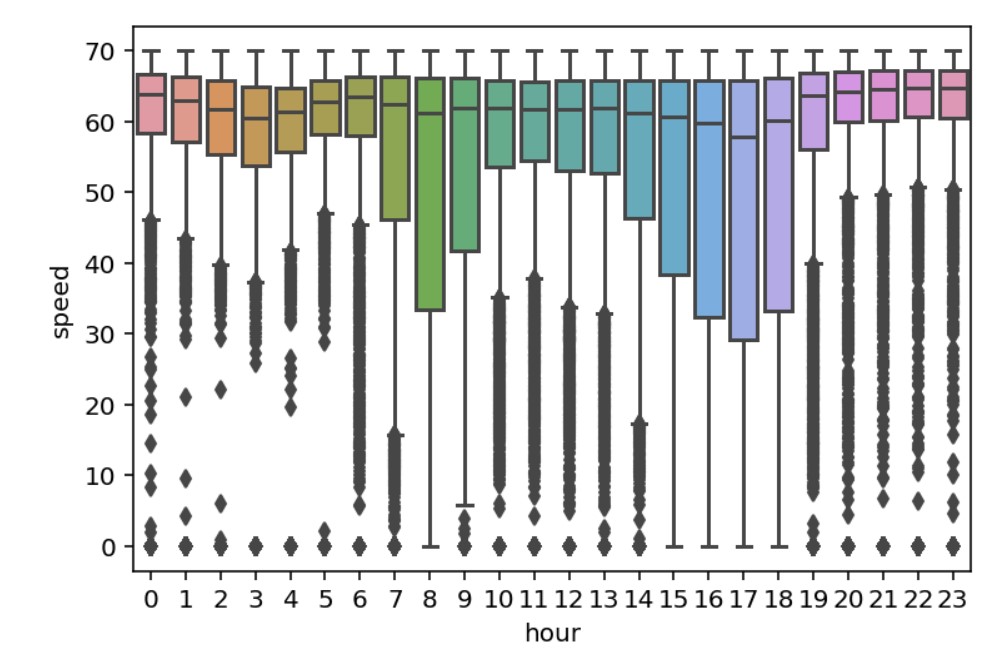
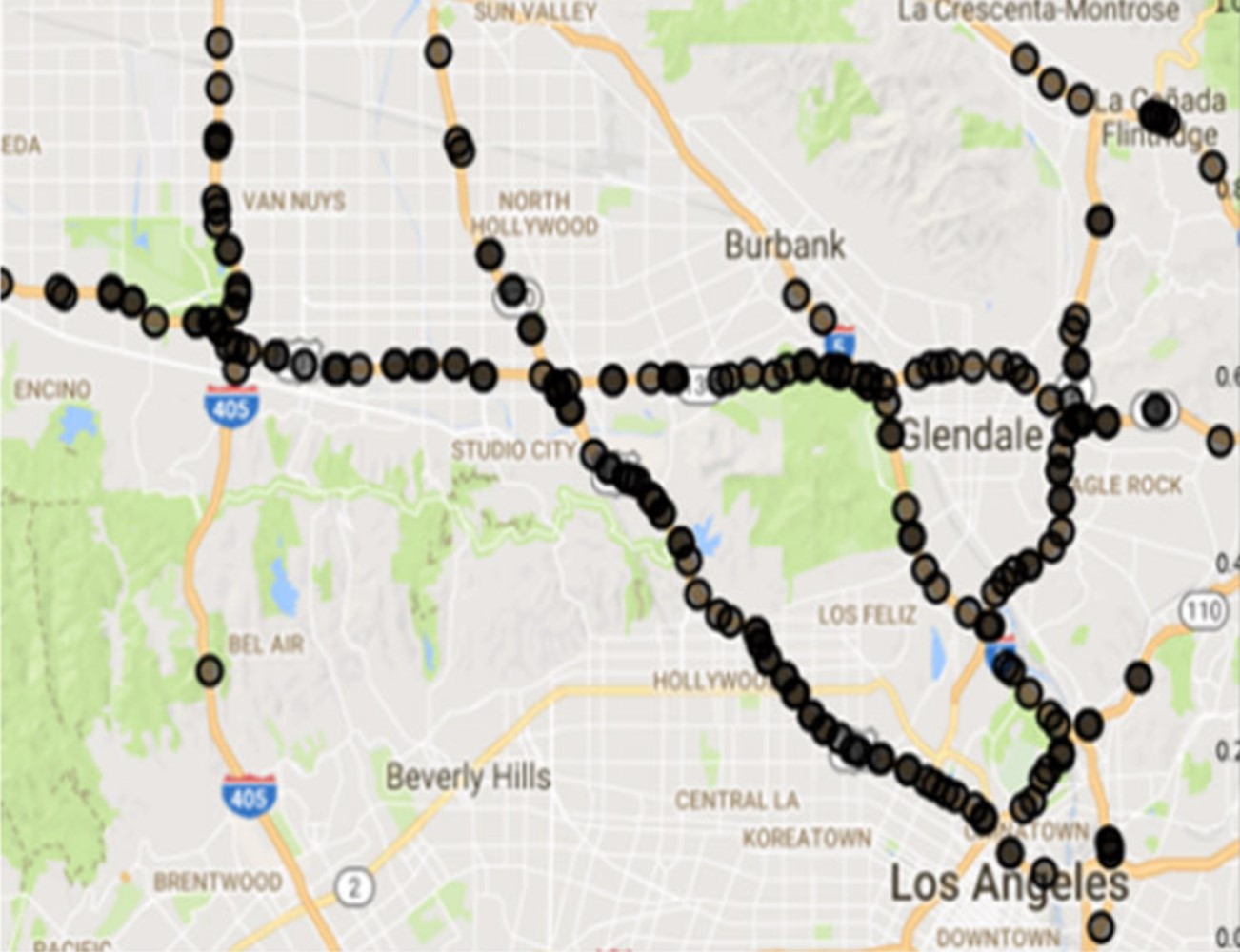
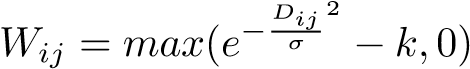


Рисунок 2: Скорость в зависимости от время суток в Лос-Анджелесе по всем

Иллюстрация 1: Распределение датчиков METR-LA

Датчики. T raffic замедляется примерно в утренние и вечерние часы пик.

Набор данных также включает матрицу направленного соседства, которая отражает силу соединения на основе расстояния от одного датчика до другого в метрах. Конкретно:

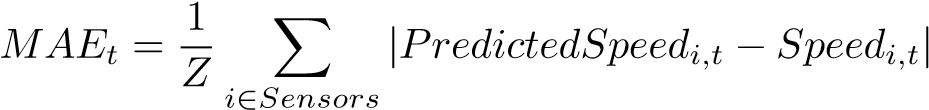


где *Wij* — вес ребра между узлами *i* и *j*, *а Dij* — расстояние в пути. Нормализующий член *σ* — это стандартное отклонение всех расстояний между любыми двумя узлами на графе (стандартное отклонение *D*), а *k* — некоторый порог, ниже которого все веса устанавливаются равными 0. В комплект входят все петли для себя. В статьях DCRNN и Graph WaveNet *k* = 0*.*1, которым мы также пользуемся.

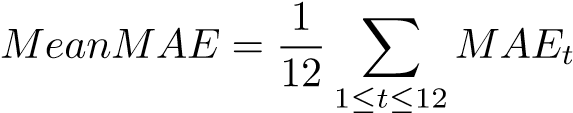
Признаки для каждого узла на заданном временном шаге имеют форму (2, 12), соответствующую скорости и времени суток для каждого из 12 временных шагов, предшествующих интересующему измерению. Поскольку для выполнения свертки графа модель должна видеть все узлы в сети, формируются пакеты (BatchSize=64, NumNodes, 12, 2).

# МЕТРИКА ОЦЕНКИ

Как и в статье WaveNet (Oord et al., 2016), мы используем среднюю абсолютную ошибку (MAE) как потери, которые модель распространяет в обратном направлении, так и сообщаемую метрику:



где *Z* = |Датчики| — нормализующая константа, а *t* — горизонт прогнозирования. Мы также продолжаем дружественное соглашение о присвоении 0 потерь примерно 5% наблюдениям, где *Speedi,t* = 0, которые, по мнению авторов DCRNN, представляют собой интервалы, когда ни один автомобиль не проезжал через датчик, а не остановившееся движение. Метрика MeanMAE



представляет собой среднее значение MAE по всем горизонтам прогнозирования. Обратите внимание, что каждое *t* имеет приращение в 5 минут, поэтому горизонт *t* = 12 — это один час вперед.

# СВЯЗАННАЯ РАБОТА

Недавно были опубликованы две статьи в области прогнозирования трафика, работу над которыми мы активно используем. DCRNN (Li et al., 2018) является первой работой, в которой используется свертка графа на графе дорожной сети, который они разделили на две матрицы смежности: для нисходящего трафика и () для восходящего трафика. Затем они вычисляют *операцию диффузионной свертки*, показанную в уравнении (2). Graph WaveNet принимает это разделение, и мы его не изменяем.

DCRNN использует стробированную рекуррентную единицу (GRU) для моделирования краткосрочных временных взаимодействий и значительно превосходит всех предшественников, которые либо не являются нейронными, либо не используют все преимущества структуры сети. Граф WaveNet (Wu et al., 2019) устраняет два недостатка DCRNN. Во-первых, матрица соседства на основе расстояний, построенная DCRNN, предполагает, что датчики всегда коррелируют друг с другом, когда они пространственно близки. Это гораздо более выгодный априорный подход, чем обучение по случайной матрице смежности, но может преувеличивать важность границы между двумя ближайшими датчиками, даже если маршрут между ними вряд ли будет пройден. Например, если два датчика находятся на соседних дорогах с односторонним движением, идущих в противоположных направлениях, оба *датчика Ei,j* и *Ej,i* будут большими в матрице смежности, даже несмотря на то, что трафик в точке *j* не очень хорошо предсказывает трафик в точке *i* и наоборот.

Graph WaveNet (Wu et al., 2019) пытается ослабить это предположение, изучая исходные и целевые вложения размера *d* = 10 для каждого датчика и используя:

*Wlearned* = *SoftMax*(*ReLU*(*matmul*(*embsrc,embdest*)))

в дополнение к фиксированным матрицам смежности, основанным на расстоянии.

DCRNN также использовала очень дорогую настройку декодера энкодера, обучение которой заняло много времени. GWN заменяет настройку кодировщика на основе GRU на блоки, вдохновленные оригинальным WaveNet (Oord et al., 2016), который впервые использовался для генерации звука. Эти блоки, показанные на рисунке 3, используют одномерные и двумерные расширенные свертки и предсказывают все 12 временных шагов одновременно, а не декодируют по одному шагу за раз. В совокупности эти изменения сокращают время обучения в 6 раз, а время вывода — в 10 раз. Они также помогают производительности: GWN улучшается по сравнению с DCRNN в MeanMAE на 0,07 (с 3,11 до 3,04).

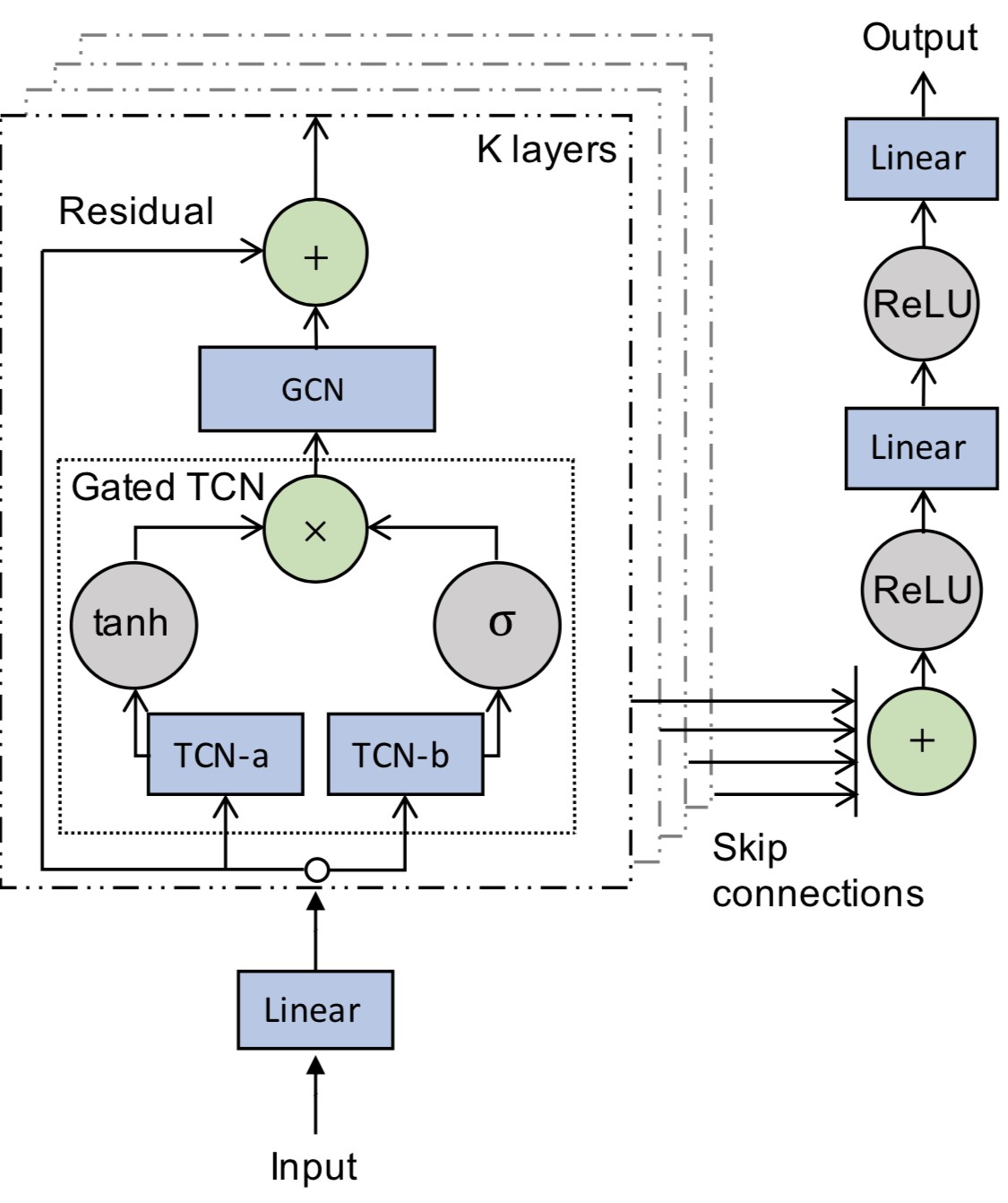


Рисунок 3: Диаграмма блоков, вдохновленных WaveNet, от GWN. Архитектура заимствована у Graph WaveNet. Стробированный TCN является одномерным временным сверточным модулем, в то время как блок GCN включает в себя изученную матрицу смежности.

# УЛУЧШЕНИЯ

## ГИПЕРПАРАМЕТРЫ

Мы скорректировали несколько гиперпараметров во всех наших моделях и наблюдаем падение MAE на 0,03 по сравнению с исходным графиком WaveNet. Влияние каждого отдельного изменения гиперпараметра показано в таблице 2.

Снижение скорости обучения: Мы умножаем скорость обучения на 0,97 после каждой эпохи. График WaveNet не использовал затухание скорости обучения, эффективно умножая на 1 после каждой эпохи.

Количество фильтров: Мы обнаружили, что увеличение числа фильтров, используемых всеми слоями в блоке Gated-TCN и блоке GCN с 32 до 40, повышает производительность при увеличении времени обучения всего на 5%. Это изменение увеличивает количество обучаемых параметров с 309 400 до 477 872 (54%).

Обрезка градиента: Мы обнаружили, что обрезка градиентов до *L2 = 3* дает меньшую ошибку, чем исходный Graph WaveNet, который обрезал градиенты до *L2 = 5*.

Представление отсутствующих данных: Как упоминалось ранее, функция потерь присваивает 0 потерь наблюдениям, где показания датчика равны 0, что означает, что ни один автомобиль не прошел мимо датчика в течение 5-минутного интервала. Примерно 5% данных содержат эти нули, и хотя они не вызывают потерь, когда они находятся в объектах, они все же вносят бессмысленные числа в прогнозы ближайших датчиков, когда они находятся в объектах. С помощью этого представления модель должна узнать, что измерения более низкой скорости указывают на более высокий трафик, за исключением случаев, когда измерение скорости равно 0, и в этом случае оно указывает на отсутствие трафика. Поэтому мы заменяем эти 0 на среднюю скорость в обучающих данных и получаем еще одно улучшение MAE на 0,01, а также более быструю сходимость.

## БОЛЬШЕ СТЫКОВОК SKIP

Из рисунка 4 видно, что выход стробированной TCN на каждом слое проходит через GCN напрямую, и промежуточные результаты не переносятся независимо от него. В результате, во время обратного распространения, Graph WaveNet пропускает очень малые градиенты обратно к слоям ранней свертки, которые наиболее удалены от расчета потерь. Чтобы исправить это, в каждом блоке мы добавляем выход стробированной TCN к выходу GCN в каждом блоке, эффективно обеспечивая пропускное соединение вокруг GCN, аналогичное пропускающим соединениям, которые уже существуют между целыми слоями. Перефразируя, где ранее вычислялся каждый слой

*xi+1* = *GraphConv*(*ha*)*,*

Теперь мы вычисляем

*xi+1* = *га* + *GraphConv*(*га*)*,*

Тем самым градиенты имеют более прямой путь к самым ранним слоям сети. Это изменение увеличивает градиенты в ранних блоках, что усиливает необходимость более строгой обрезки градиентов. В нашей модели градиенты обрезаются при *L2norm* = 3, что является уменьшением от *L2norm* = 5 в исходном GWN.

## ПРЕТРЕНИНГ НА БОЛЕЕ КОРОТКИХ ГОРИЗОНТАХ ПРОГНОЗИРОВАНИЯ

Мы обнаружили, что модели, обученные на краткосрочном подмножестве полного горизонта прогнозирования за 1 час, работают лучше в этих краткосрочных прогнозах, чем модели, обученные прогнозировать полный час, как показано на рисунке 4. Другими словами, модели, которым позволено специализироваться на краткосрочном трафике, лучше справляются с краткосрочными прогнозами, чем те, которые пытаются изучить краткосрочные и долгосрочные закономерности одновременно.

Это открытие побудило нас инициализировать веса для полной задачи весами, изученными на задачах на более коротких дистанциях, что обеспечило снижение MAE на 0,01 (усредненное за 2 прогона). Этот подход с предварительным обучением и тонкой настройкой занимает всего на 7 минут больше, чем обучение с нуля, потому что модель короткого горизонта сходится через 60 эпох, а тонко настроенная модель сходится всего через 29 эпох. Обучение на полной задаче с нуля сходится через 100 эпох.2

2

80 минут на графическом процессоре NVIDIA v100. И, каким-то образом, 4 часа на графическом процессоре T4!

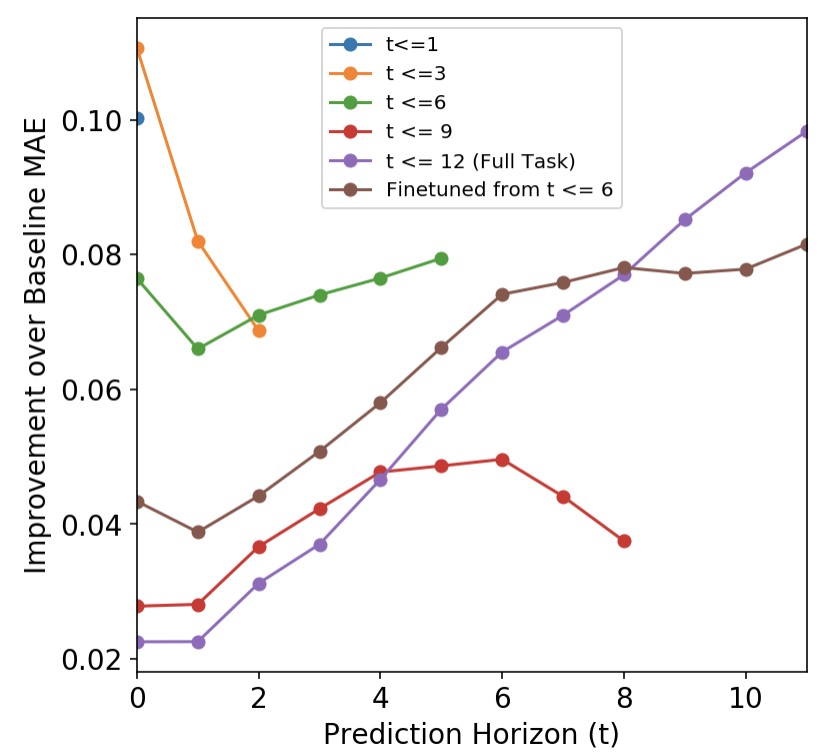


Рисунок 4: Улучшение MAE по сравнению с базовым уровнем для различных задач предварительного обучения и тонкой настройки. Например, *t* ≤ 3 — это модель, обученная прогнозировать на 3 временных шага (5, 10 и 15 минут) вперед, и, следовательно, имеет только 3 точки данных на диаграмме. Мы видим, что модели, обученные прогнозировать трафик на короткие расстояния, делают это лучше, чем модели, обученные прогнозировать как на краткосрочные, так и на дальние расстояния. Мы строим график улучшения по сравнению с базовым уровнем, а не с абсолютным MAE, потому что MAE на более поздних временных шагах значительно увеличивается, что затрудняет визуализацию различий между моделями.

По мере того как модель улучшается на более длинных горизонтах, она ухудшается на более коротких. На рисунке 4 коричневая кривая отстает от зеленой кривой (на основе которой она была точно настроена) на коротких горизонтах. По этой причине, использование ансамбля зеленой модели короткого горизонта (для прогнозирования краткосрочного горизонта) и коричневой тонко настроенной модели (для прогнозирования более длительного горизонта) обеспечивает дальнейшее снижение MeanMAE на 0,01 без дополнительных затрат на обучение. Мы не тратили время на оптимизацию гиперпараметров для тонкой настройки или сборки и могут быть модификации, которые уменьшают «забывчивость» кратковременной предобучающей задачи.

Мы предполагаем, что более короткие диапазоны прогнозирования позволяют модели игнорировать более удаленные узлы и сосредоточиться на более простой задаче, но не все наши экспериментальные данные подтверждают это. Например, модель, обученная прогнозировать только 7 ≤ *т* ≤ 12 (не показана на графике), показала гораздо худшие результаты, чем модель, обученная на полной задаче: 1 ≤ *т* ≤ 12, хотя она должна быть в состоянии «сфокусироваться» больше. Во-вторых, модель, обученная прогнозировать только следующие 45 минут (красный), показала худшие результаты по данным проверки, чем модель, обученная прогнозировать следующий час (фиолетовый). Эта динамика заслуживает дальнейшего изучения.

# РЕЗУЛЬТАТЫ

## ОБЗОР

Используя комбинацию небольших изменений в архитектуре Graph WaveNet и расписании обучения, мы уменьшили погрешность этой модели в задаче METR-LA с 3,04 до 2,98, что является новым современным решением этой задачи прогнозирования трафика. Обзор улучшений, достигнутых в результате каждой из модификаций, можно найти в таблице 1.

Следуя прецеденту, созданному DCRNN и Graph WaveNet, мы разделили данные на 70%, 10% и 20% для нашего обучения, валидации и тестирования соответственно. Разбиения отражают время: набор поездов предшествует проверочному набору, который предшествует тестовому набору. Мы рассмотрели только валидационный MAE для выбора модели и ранней остановки, но мы сообщаем о тестовом MAE в таблицах и рисунках, чтобы облегчить сравнение с документом GWN, в котором не сообщается о результатах валидации. Во всех наших прогонах MAE тестового набора на 94% коррелировал с проверочным MAE, а улучшения, которые мы обсуждаем, улучшили как валидационный MAE, так и тестовый MAE. Приняв решение об окончательном наборе подходов, мы проверили все в PEMS-BAY, чтобы убедиться, что наши изменения обобщаются на другой набор данных о трафике.

МЕТР-ЛА

ЗАЛИВ ПЕМСА

Мин

Значить

30

Модель

15

Мин

Мин

60

Значить

30

Мин

60

15

Мин

Мин

МАЕ

МАЕ

3.15

2.77

3.60

DCRNN (Сообщается)

3.11

1.680

1.380

1.740

2.070

3.07

3.04

GWN (Сообщается)

2.69

3.53

1.58

1.30

1.63

1.95

GWN (реплицированный)

2.70

3.10

3.55

3.06

1.591

1.316

1.647

1.968

GWNV2 (наш)

2.67

3.04

3.45

3.00

1.303

1.622

1.909

1.560

Тонкая настройка (наша)

2.66

3.03

3.47

2.99

1.299

1.613

1.896

1.552

RangeEnsemble (Наш)

2.64

3.03

3.47

2.98

1.286

1.606

1.896

1.546

Таблица 1: МАЭ для разных моделей. GWNV2 включает в себя соединение skip вокруг GCN, а также модифицированные гиперпараметры (более подробную информацию о том, как каждая модификация влияет на это число, можно найти в таблице 2). Тонко настроенная модель инициализируется весами модели, обученной прогнозировать трафик на 30 минут вперед. RangeEnsemble использует ту же 30-минутную модель для прогнозирования первых 30 минут и тонко настроенную модель для прогнозирования 35–60 минут. Все сообщаемые метрики являются средним значением 2 запусков.

# АНАЛИЗ

## НЕУДАЧНЫЕ ЭКСПЕРИМЕНТЫ

Чтобы сэкономить кредиты других исследователей на облачные вычисления и частично компенсировать наше воздействие на выбросы углекислого газа, мы кратко перечислим эксперименты, которые мы пробовали, но которые не улучшили валидацию MAE.

1. Добавление информации о дне недели.
   * Опытным путем, скорость движения примерно на 10 миль в час ниже в будние дни в 5 часов вечера, чем в выходные. Однако добавление дня недели в модель в виде одного горячего представления или в виде скаляра не повлияло на метрики и даже не привело к более быстрой сходимости модели. Либо эту информацию нужно добавить в другом формате, либо это указывает на то, что краткосрочные показания трафика уже адекватно отражают эти тенденции. Интуитивно кажется, что должно быть несколько минут, прямо перед началом часа пик, когда в последнее время очень мало движения, и знание дня недели поможет предсказать, будет ли час пик. Также может иметь смысл использовать бизнес-календарь для создания функции двоичного файла isWorkday.
2. Использование последних 75 минут вместо последних 60.
3. Изменение функций отсева, скорости обучения, снижения веса, активации.
4. Цикличность темпов обучения.
   * Циклические скорости обучения (Smith, 2017) приводят к более быстрому обучению и лучшим показателям во многих контекстах, но не в этом!
5. Передача знаний между дорожными сетями
   * Мы попробовали тонкую настройку модели на METR-LA на основе полностью обученной контрольной точки на PEMSBAY (помимо смежных матриц, которые имеют неправильную форму, а также явно не должны переноситься). Эта модель была примерно на 0,03 MAE хуже, чем обучение с нуля, хотя она обучалась быстрее, что говорит о достижении некоторого локального минимума. Это также не очень полезно на практике, поскольку нам известно только два таких набора данных о трафике.
6. Большие размеры партий и половинная точность.
   * Переход на половинную точность на самом деле немного замедлил тренировку и значительно повредил метрикам. При стандартном размере пакета модель не является узким местом памяти, требуя всего 4 ГБ оперативной памяти графического процессора. Мы постарались увеличить размер партии как с половинной точностью, так и без нее. В обоих случаях больший размер пакета отрицательно влиял на производительность (даже при соразмерном увеличении скорости обучения), а умножение размера пакета на 8 ускоряло обучение только на 10%.
7. Использование трансформаторов вместо 1D сверток.
   * Пропускание признаков через 1 трансформатор с 1 головой внимания в самом начале модели навредило метрикам. Добавление любого трансформатора внутри блока, которых здесь 8, делало обучение невыносимо медленным. Мы пропустили пробежку, и она была очень плохой, но слишком медленной и недостаточно многообещающей, чтобы продолжать итерацию.
8. Удаление удаления информации в блоке.
   * Каждый слой уменьшает временную размерность тензора на 1 или 2, поэтому пропуск соединений предыдущих слоев отбрасывает выходы для «дополнительных временных шагов» из самой ранней части предыдущего выхода. Мы попытались решить эту проблему, объединив по средним значениям каналы, которые были бы отброшены. Это не улучшило производительность и отражает тенденцию к тому, что последние 30 минут гораздо полезнее для прогнозирования, чем следующие 30.
9. Инициализация изученной матрицы смежности из разложения СВД предоставленной дорожной сети.
10. Прямое обратное распространение RMSE вместо MAE немного улучшает RMSE, но вредит MAE.
11. Изменение температуры софтмакса, применяемого к изученной матрице смежности.
12. Изменение количества слоев в блоке. (Это сильно замедляет процесс!)
13. Уменьшение/увеличение размера вложений изученных узлов, *embsrc* и *embdest*.
14. Добавление пакетной нормализации после шага свертки графа.

## АБЛЯЦИИ

В таблице 2 показано, что ни одно из наших изменений не может быть удалено без снижения производительности. В частности, снижение скорости обучения, простое изменение, имеет важное значение.

|  |  |
| --- | --- |
| Модификация | Среднее значение MAE |
| Базовый уровень GWN (без изменений) | 3.057 |
| без n каналов=40 | 3.024 |
| без подключения к пропуску | 3.013 |
| без 0 замены | 3.010 |
| без обрезки града=3 | 3.023 |
| Без распада LR | 3.032 |
| со всеми модификациями | 3.002 |

Таблица 2: Мы экспериментируем с удалением каждого из предложенных нами изменений из окончательной архитектуры и сбросом их к значениям по умолчанию. Значения по умолчанию: clip=5, lr decay=1 (без затухания), n каналов=32.

Мы также предоставляем абляцию по количеству использованных временных шагов. Мы обнаружили, что использование большего количества истории для составления прогнозов, как правило, помогает, но улучшение не является монотонным и начинает стабилизироваться после 5 временных шагов. В таблице ниже мы показываем производительность моделей, которым предоставлен доступ к разным объемам истории. Все горизонты прогнозирования — это 60 минут (12 временных шагов) в будущее.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Длина истории | 1 | 2 | 3 | 4 | 5 | 6 | 9 | 12 |
| МАЕ | 3.058 | 3.034 | 3.014 | 3.017 | 3.007 | 3.015 | 3.024 | 3.002 |

Таблица 3: Среднее значение MAE моделей, получивших доступ к различным объемам истории. Например, History Length=6 использует последние 30 минут. Все модели обучены прогнозировать следующий час.

Наконец, мы убедились, что свертка графа в целом и изученная матрица смежности Graph WaveNet имеют решающее значение для высокой производительности. Без свертки графика (или других модификаций) мы получаем среднее MAE 3.59. Без изученной матрицы смежности мы получаем среднее значение MAE, равное 3,08. И, без учета информации о времени суток или модификаций, модель достигает MeanMAE 3,1. Этот последний результат на удивление хорош и подтверждает общую тенденцию, которую мы наблюдали, что краткосрочная информация о дорожном движении гораздо полезнее, чем долгосрочная информация и атрибуты, такие как время суток, день недели или показания трафика более чем за час назад.

# ЗАКЛЮЧЕНИЕ И БУДУЩАЯ РАБОТА

Мы показали, что несколько небольших изменений в архитектуре Graph WaveNet: увеличение размера некоторых внутренних сверточных слоев, пропуск соединения, снижение скорости обучения и изменение представления отсутствующих данных улучшают производительность на обоих наборах данных трафика, с которыми мы экспериментировали. Замена нулей в функциях на среднюю скорость в улучшенных результатах, но мы считаем, что немного более сложные схемы интерполяции, такие как копирование самых последних ненулевых показаний датчика, могут дать дополнительные преимущества.

Мы также показываем, что предварительное обучение на более высоких скоростях краткосрочного трафика улучшает полную производительность задачи, но по мере того, как тонко настроенная модель изучает отношения на более длинных горизонтах, она теряет часть своей производительности на более коротких горизонтах. Загадочным образом модель, обученная прогнозировать только долгосрочные горизонты трафика, показала плохие результаты. Будущая работа должна быть направлена на разгадку этой тайны и, в идеале, на поиск способа обучения одной модели, которая превосходна как в краткосрочном, так и в долгосрочном прогнозировании трафика.

# ССЫЛКИ

Ягуан Ли, Роуз Юй, Сайрус Шахаби и Янь Лю. Диффузионная сверточная рекуррентная нейронная сеть: прогнозирование трафика на основе данных. В *Международной конференции по учебным представлениям (ICLR '18),* 2018.

Аарон ван ден Орд, Сандер Дилеман, Хейга Зен, Карен Симонян, Ориол Виньялс, Алекс Грейвс, Нал Калчбреннер, Эндрю Сениор и Корай Кавукчуоглу. Wavenet: генеративная модель для необработанного звука. *Препринт arXiv arXiv:1609.03499*, 2016.

Лесли Н. Смит. Циклические скорости обучения для обучения нейронных сетей. *Зимняя конференция IEEE по приложениям компьютерного зрения (WACV) 2017 г*., март 2017 г. DOI: 10.1109/wacv.2017.58. URL [http:](http://dx.doi.org/10.1109/WACV.2017.58)

[dx.doi.org/10.1109/WACV.2017.58.](http://dx.doi.org/10.1109/WACV.2017.58)

Цзунхань Ву, Шируй Пань, Годун Лун, Цзин Цзян и Чэнци Чжан. Графовый вейвнет для глубокого пространственно-временного моделирования графов. *Препринт arXiv arXiv:1906.00121*, 2019.