

# Автоматизированное построение классификаторов на основе рекуррентных нейронных сетей для детекции проездов транспортных средств

Евгений Бурнаев  
ИППИ РАН, Сколтех  
burnaev@iitp.ru

Иван Коптелов  
ИППИ РАН  
koptelov@iitp.ru

Герман Новиков  
ИППИ РАН  
novikov@iitp.ru

Тимур Ханипов  
ИППИ РАН  
timur.khanipov@iitp.ru

## Аннотация

*Рекуррентные нейронные сети (RNN) широко применяются для прогнозирования временных рядов. В данной статье мы используем LSTM-RNN для автоматизированного создания бинарного детектора проездов транспортных средств по многомерному, зависящему от времени вектору сигналов датчиков с пункта пропуска автомобилей. Полученные результаты позволяют сделать выводы о том, что человеческий труд по созданию классификатора, состоящего из набора детерминированных правил, можно успешно заменить на обученную по размеченным данным RNN.*

*Ключевые слова: рекуррентные нейронные сети, классификация, временные ряды.*

## 1. Введение

В [1] описывается автоматический классификатор транспортных средств (АКТС) для платных дорог на основе видеоклассификации, используемый на большинстве платных дорог России. Одним из важнейших модулей АКТС является детектор проездов, определяющий факт наличия транспортного средства (ТС). Детектор проездов по набору из нескольких бинарных сигналов, получаемых от нижестоящих подсистем (бинарных детекторов), принимает решение о факте проезда (наличие ТС в зоне классификации). Базовая логика работы детектора проездов основана на “голосуемой схеме”, в рамках которой наличие проезда фиксируется, если большинство бинарных детекторов дает положительный ответ. Эта логика дополняется рядом добавленных

экспертом-разработчиком эмпирических правил, в которых анализируются временные задержки между переключениями бинарных сигналов, последовательности переключений и другая информация. Эти правила изменялись и дополнялись в течение опытно-промышленной эксплуатации АКТС по результатам анализа допущенных ошибок.

В предыдущей статье, посвященной детектированию проездов в АКТС [2], было указано качество детектирования ТС, равное 99.58%. За прошедшее время тестовый стенд, используемый для отладки классификатора ТС, был дополнен новыми данными с проездов, на которых происходили ошибки детектирования и классификации ТС. Вдобавок, в текущей статье используются результаты прогона системы с отключенной функцией детектирования сцепных устройств. Качество работы версии системы классификации ТС из прошлой статьи с отключенным детектором сцепок на новом тестовом стенде составляет 83.47%. На современной версии классификатора без детектирования сцепок качество составляет 88.90%, а с детектированием — 91.10%. По сравнению со старой версией, в новой версии системы классификации ТС были оптимизированы алгоритмы работы детектора щита и сцепок, алгоритмы работы корреляционного детектора, а также был улучшен алгоритм комбинирования показаний детекторов.

Создание такого рода правил является достаточно трудоемкой работой, требующей творческого подхода. Представляет интерес автоматизированный подход, при котором вместо человека экспертом выступает технология машинного обучения. Потенциально это может позволить получить более высокое качество классификации, нежели даваемое эм-

пирическими правилами.

Таким образом, в данной работе решается задача разработки метода, который бы позволил автоматизировать синтез классификатора ТС и минимизировать время работы специалистов.

## 2. Описание данных

Данные состоят из записей логов системы автоматической классификации транспортных средств. В каждом файле записан один или несколько проездов ТС. Все численные эксперименты проводятся на выборке реальных данных, состоящей из 4761 файла с логами системы.

В лог системы записываются (с определенной частотой дискретизации) показания трехмерного сигнала  $X_t$ , каждая компонента которого предварительно бинаризована и порождается одним из датчиков: корреляционный детектор (срабатывает при изменении изображения в видеопотоке), индукционная петля (вмонтирована в покрытие дороги и срабатывает при наличии на дороге большого количества металла), щитовой датчик (срабатывает, если на изображении заслонен щит, находящийся напротив камеры); помимо этих сигналов также записывается номер кадра, референсный сигнал — разметка, которая создается экспертом вручную при просмотре видеозаписи, и предсказания базового классификатора, основанного на построенных вручную правилах [1].

Запись значений сигналов в файлы происходит только в том случае, если состояние хотя бы одного из сигналов изменяется, иначе значения сигналов просто не записываются.

Frame No.	Shield	Loop	Cor	Base clf	Ref. pass
196	1	0	0	0	0
201	1	1	0	0	0
202	0	1	1	1	0
208	1	1	1	1	1
246	0	1	1	1	1
266	1	1	1	1	0
268	1	1	0	1	0
269	1	0	0	0	0
270	0	0	0	0	0

Таблица 1. Пример лога с разметкой

В **Таблице 1** представлен пример одного из логов. Используемые обозначения: **Frame No.** — нумерация кадров внутри каждого лога, **Shield** — щитовой датчик, **Loop** — индукционная петля, **Cor** — корреляционный датчик, **Base clf** — предсказание базового классификатора, **Ref. pass** — референсный проезд.

## 3. Критерии качества

В силу особенностей задачи для оценки качества детектирования проездов автомобилями необходимо использовать специализированный критерий, который только в предельных случаях совпадает со стандартным поточечным критерием качества двухклассовой классификации **Accuracy**. Дело в том, что стандартные критерии, в которых используются поточечные разности референсного и предсказанного сигналов, не способны оценить качество классификации проездов с физической точки зрения. Опишем используемую в данной работе критерий качества **Pass Quality (PQ)**:

$$PQ = \frac{R}{R + \sum Err}, \quad (1)$$

где  $R$  — количество верно распознанных проездов, а  $\sum Err$  вычисляется как сумма штрафов за ошибки классификации на всем наборе данных. Подсчет штрафа за каждый из видов ошибки (пропуск транспортного средства, склейка двух подряд идущих ТС и т.п.) — сложная процедура, которая описана в статье [3], см. также **Таблицу 2**.

Реф. проезд	Найденный проезд	Тип ошибки	Вес ошибки
1	1	ошибки нет	0
1	0	пропуск проезда	1
0	1	ложный проезд	1
L	1	склейка проездов	L
1	K	разрыв проезда	K
L	K	множественная ошибка	max(L, K)

Таблица 2. Штрафы за ошибки при классификации

Здесь  $L$  обозначает фактическое число проездов в рассматриваемой последовательности сигналов,  $K$  — соответствующее число проездов, детектированное в этой последовательности (подробности см. в [3]).

Данный критерий качества не учитывает то, насколько сдвинут детектированный проезд от идеального. Однако, как показывают результаты реальных экспериментов, для приложений это и не нужно, поскольку первостепенное значение имеет правильная последовательность детектированных проездов и их пересечение с реальными проездами хотя бы в один момент времени.

## 4. Разведочный анализ

В данном разделе сравниваются результаты, получаемые с помощью различных методов машинного обучения: градиентный бустинг над деревьями **XGB** (см. описание алгоритма в [4]; использовалась реализация [6]), логистическая регрессия **LR** из пакета **scikit-learn** [7], полносвязная нейронная сеть с одним скрытым слоем из 12 нейронов **NN** (см. подходы к обучению такой сети в [8, 9, 10, 11]), простая рекуррентная нейронная сеть **SimpleRNN** из пакета **Keras** [5].

В силу особенностей задачи, разбиение данных на обучающую и контрольную выборки было организовано следующим образом: логи, которые попадают в соответствующую подвыборку (обучающую или контрольную) и их порядок выбирались случайным образом, а порядок кадров внутри каждого из логов оставался неизменным.

Результаты обучения классификаторов на исходном сигнале  $X_t$  без учета его значений в предыдущие моменты времени приведены в **Таблице 3**, см. строки **XGB<sub>1</sub>**, **LR<sub>1</sub>**, **NN<sub>1</sub>**. Как следует из полученных результатов, в силу автокоррелированности компонент  $X_t$ , классификация стандартными методами, не учитывающими зависимость  $X_t$  от  $X_{t-1}, \dots, X_{t-k}, \dots$ , имеет в смысле рассматриваемого критерия качества **PQ** низкую точность. При этом совпадение результатов легко объясняется тем, что в каждый момент времени всего может быть восемь различных значений трехмерного вектора  $X_t$  (всевозможные комбинации из трёх нулей и единиц), для каждого из которых каждый из классификаторов просто строит схожие решающие правила.

Первая идея, которая позволяет улучшить точность классификации в смысле используемого критерия качества, состоит в расширении пространства признаков  $X_t$  за счет использования показаний датчиков, отстающих на  $\{1, \dots, w\}$  отсчетов. Результаты проведенных экспериментов представлены в **Таблице 3**, см. строки **XGB<sub>2</sub>**, **LR<sub>2</sub>**, **NN<sub>2</sub>**. Для каждого из классификаторов оптимальный размер окна  $w$  выбирался на основе скользящего контроля. Для логистической регрессии и нейронной сети порог получения бинаризованного сигнала из вероятности подбирался с помощью максимизации целевой функции на обучающей выборке. Видно, что расширение пространства признаков позволяет получить значимое улучшение качества. При этом наилучший результат получается у **NN<sub>2</sub>**. Однако, этот результат не превосходит результат базового классификатора.

Вторая естественная идея состоит в использовании рекуррентных нейронных сетей, которые хорошо зарекомендовали себя в задачах разметки последовательностей и классификации временных рядов. Будем использовать стандартную рекуррент-

ную сеть **SimpleRNN** с одним рекуррентным и одним скрытым слоями [12]. Оказывается, что данная модель позволяет получить достаточно хорошее качество, наиболее близкое к качеству базового классификатора.

Классификатор	R	$\sum Err$	PQ
<b>XGB<sub>1</sub></b>	<b>3090.2</b>	2802.3	0.532
<b>LR<sub>1</sub></b>	<b>3090.2</b>	2802.3	0.532
<b>NN<sub>1</sub></b>	<b>3090.2</b>	2802.3	0.532
<b>XGB<sub>2</sub></b>	1806.0	453.0	0.799
<b>LR<sub>2</sub></b>	1794.1	302.6	0.856
<b>NN<sub>2</sub></b>	1758.0	276.6	0.864
<b>SimpleRNN</b>	1784.8	<b>214.3</b>	<b>0.892</b>
<b>Base Classifier</b>	1684.3	158.7	0.914

**Таблица 3. Сравнение различных моделей классификаторов**

**Замечание:** в базовом классификаторе **BaseClassifier** использовался дополнительный признак, сигнализирующий о “сцепке” проездов. Этот признак позволяет лучше различать идущие на малом расстоянии автомобили и автомобили с прицепами, и тем самым улучшает базовый результат в смысле значения критерия качества **PQ** с **0.889** до **0.914**. В экспериментах, описанных в данной работе, указанный дополнительный признак не учитывался, хотя его использование потенциально позволит еще больше увеличить качество классификации.

Основная причина низкого качества предсказаний классификаторов состоит в том, что во всех использованных алгоритмах оптимизируется отличная от целевого критерия качества величина — среднеквадратическая ошибка прогноза **MSE**, которая не достаточно хорошо согласуется с критерием качества **PQ** (ошибкой мы называем величину  $PQE = 1 - PQ$ ).

## 5. Классификация на основе рекуррентных нейронных сетей

Проведенное исследование с использованием различных классификаторов показывает, что к результатам базового классификатора, построенного вручную методом подбора правил, распределенных во времени, получается приблизиться наилучшим образом используя модель рекуррентной нейронной сети. Таким образом, использование моделей рекуррентных нейронных сетей является потенциально перспективным направлением автоматизации построения классификаторов и дальнейшего повышения точности детектирования проездов.

## 5.1. Архитектуры рекуррентных сетей

Классическая модель рекуррентной нейронной сети **SimpleRNN** представляет собой нейронную сеть с одним скрытым слоем, в котором выходной сигнал каждого отдельного нейрона подается ему же на вход в следующий момент времени. В случае сигналов значительной продолжительности при обучении модели и подсчете градиентов функции качества обучения возникает эффект “взрывающихся и исчезающих градиентов” (*the exploding and the vanishing gradient problems* [15]). Этот эффект связан с тем, что градиенты функции качества обучения зависят от произведения величин градиентов нейронов в скрытом слое, оцененных для всех последовательных значений сигналов в обучающей выборке, см. рис. 1; как следствие, это произведение может принимать как значительные, так и малые по абсолютной величине значения.

Одним из способов избежать появления этого эффекта является использование архитектуры нейронной сети Long Short-Term Memory (**LSTM**), которая позволяет эффективно моделировать зависимости с длинной памятью в сигналах.

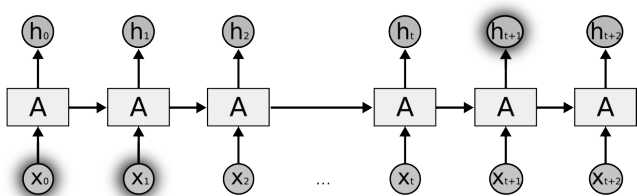


Рис. 1. Долговременная зависимость в RNN

В 2014 году в работе [16] была представлена модель архитектуры нейронной сети Gated Recurrent Unit (**GRU**), основанная на тех же принципах, что и **LSTM**, но использующая более экономичную параметризацию и меньшее число операций для вычисления выходного сигнала. Более обширный обзор архитектур **RNN** представлен в работах [13, 14].

## 5.2. Выбор архитектуры RNN

Для использования **RNN** на практике необходимо произвести поиск ее наилучшей архитектуры. В данном разделе описаны результаты экспериментов по поиску оптимальной для данной задачи архитектуры нейронной сети. В экспериментах использовались следующие типы модели **RNN**: **LSTM**, **GRU** и **SimpleRNN**. Все эксперименты проводились с использованием фреймворка **Keras** [5], который является оберткой над **Python**-библиотеками **Theano** [17] и **TensorFlow** [18].

Для каждого из типов модели **RNN** был проведен ряд экспериментов, в которых подбирались мак-

ропараметры сети, функции активации и длина окна  $w$ . В силу того, что для каждой итерации обучения новой рекуррентной сети с несколькими слоями требуется достаточно много времени, для подбора макропараметров модели каждого типа количество скрытых слоев было выбрано и зафиксировано в начале эксперимента равным одному, количество нейронов в каждом слое ограничено сверху восемью, а максимальное количество эпох в каждой итерации обучения согласно результатам предварительных экспериментов было выбрано равным 40. В **Таблице 4** отражены усредненные результаты по 30 экспериментам с лучшей моделью **RNN** каждого типа, выбранной в описанном выше пространстве макропараметров. Как видно, подбор параметров даже в ограниченном пространстве значительно улучшил качество модели **SimpleRNN** по сравнению с первыми экспериментами, представленными в **Таблице 3**. По результатам проведенных экспериментов было решено в дальнейшем использовать архитектуру с **LSTM** слоями, показавшую наилучшее качество детектирования среди сравниваемых типов моделей.

Модель	R	$\Sigma Err$	PQ
<b>SimpleRNN</b>	1723.0	161.1	0.915
<b>LSTM</b>	<b>1751.8</b>	<b>145.3</b>	<b>0.923</b>
<b>GRU</b>	1699.0	175.0	0.907
<b>Base Classifier</b>	1684.3	158.7	0.914

Таблица 4. Сравнение различных моделей RNN

## 5.3. Улучшение качества классификации

В силу того, что оптимизируемая при обучении нейронной сети среднеквадратическая ошибка не является целевой в нашей задаче, дополнительно рассматривались различные подходы к улучшению финального результата детектирования в смысле критерия качества **PQ**. Были опробованы различные подходы: использование взвешенной среднеквадратической ошибки в качестве целевой функции, используемой при обучении нейронной сети; сглаживание входных сигналов с помощью морфологического фильтра (округления по математическим правилам среднего между значениями по времени в единичной окрестности каждой точки); добавление штрафа на производную выходного сигнала нейронной сети в ее целевую функцию, используемую при обучении; оптимизация порогового значения выходного сигнала по целевому критерию качества **PQ** на обучающей выборке; применение морфологического фильтра к выходному сигналу нейронной сети.

Наилучший прирост качества детекции проездов ТС получается за счет значительного расширения структуры модели нейронной сети **LSTM**. В ре-

зультате проведенных экспериментов получена оптимальная архитектура сети, которая изображена на рис. 2: один входной LSTM-слой и два скрытых Dense слоя. Однако, если не использовать преобразования типа Dropout, несмотря на то, что среднеквадратическая ошибка MSE на валидационной выборке падает до 0.06 – 0.07, тем не менее происходит увеличение целевой ошибки PQE. В свою очередь добавление преобразования типа Dropout перед каждым Dense слоем увеличивает MSE до 0.13 понижая при этом ошибку PQE на 0.005 – 0.010.

Из всех остальных рассмотренных модификаций наиболее значимое увеличение точности детекции проездов ТС дает подбор порогового значения выходного сигнала с помощью скользящего контроля на обучающей выборке и последующее применение морфологического фильтра к бинаризованному с помощью выбранного порогового значения выходному сигналу нейронной сети.

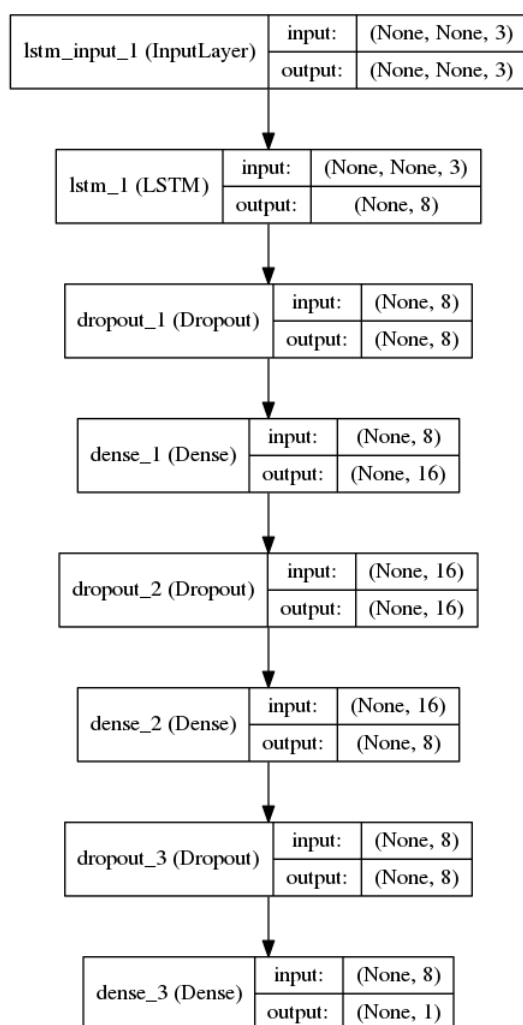


Рис. 2. Финальная модель LSTM-RNN

Также была проведена оценка важности каждо-

го из входных признаков с точки зрения его влияния на точность финальной модели. В Таблице 5 приведены значения критерия качества для моделей, построенных с использованием всевозможных комбинаций входных признаков. Из неё видно, что набор из (Shield, Cor) даёт наилучшее качество.

Признаки	R	$\sum Err$	PQ
<b>Cor</b>	1714.1	153.9	0.918
<b>Loop</b>	1621.6	255.9	0.864
<b>Shield</b>	1679.5	200.8	0.893
<b>(Loop, Cor)</b>	1713.4	170.6	0.910
<b>(Shield, Loop)</b>	1738.7	130.8	0.930
<b>(Shield, Cor)</b>	<b>1769.9</b>	<b>91.8</b>	<b>0.952</b>
<b>(Shield, Loop, Cor)</b>	1767.6	93.2	0.950
<b>Base Classifier</b>	1684.3	158.7	0.914

Таблица 5. Качество финальной модели LSTM-RNN на разных подмножествах набора входных признаков

## 6. Выводы

Итак, полученные результаты показывают заметно более высокое качество классификации, равное 0.952, с использованием только двух входных признаков, тогда как базовый классификатор для достижения качества детекции PQ, равного 0.914, использует дополнительный четвертый признак — детектор сценки, без которого качество базового алгоритма падает до 0.889.

Таким образом, данное исследование позволило получить автоматизированный алгоритм построения и обучения классификатора, который превосходит в смысле качества детекции проезда ТС полученные ранее результаты. На данном этапе продолжение исследования возможно в нескольких направлениях.

Во-первых, это повышение точности классификации за счет оптимизации критерия качества PQ напрямую при обучении нейронной сети. Реализация такого рода алгоритма обучения возможна за счет использования безградиентных методов оптимизации.

Во-вторых, это создание агрегирующего фильтра, принимающего на вход результаты предсказаний разных моделей детекции.

И, наконец, это реализация комплексного решения, в рамках которого будет полностью исключены первоначальная предобработка данных с помощью классических методов распознавания изображений и другие дополнительные шаги по обработке данных, которые происходят на этапе между съемкой автомобиля камерой и получением бинаризован-

ных входных сигналов  $X_t$ . То есть предполагается использование следующей нейросетевой структуры, объединяющей всю подсистему детектирования проездов на АКТС в стек из набора сверточных сетей:

- на первом уровне используется набор сверточных сетей, обрабатывающих изображения с каждой из камер пункта пропуска автомобилей, и выделяющих признаки из этих изображений;
- на следующих уровнях выделенные с помощью сверточных сетей признаки посредством модели типа **RNN** комбинируются с сигналами, получаемыми из индукционной петли и других устройств пункта пропуска автомобилей;
- далее, модель типа **RNN** со структурой, аналогичной изображенной на рис. 2 архитектуре нейронной сети, используется для распознавания проезда.

## Благодарности

Работа была выполнена при поддержке грантов РФФИ 16-01-00576 А и 16-29-09649 офи\_м.

## Список литературы

- [1] Т. Khanipov, I. Koptelov, A. Grigoryev, E. Kuznetsova, D. Nikolaev, *Vision-based Industrial Automatic Vehicle Classifier* // Seventh International Conference on Machine Vision (ICMV 2014), SPIE, 9445 (2015).
- [2] Д. Бочаров, И. Коптелов, Е. Кузнецова, *Детекторы проездов на основе технического зрения в автоматическом классификаторе транспортных средств* // Information Technology and Systems, 2015, pp. 485–497.
- [3] Малюгина О. В. *Оценка качества систем автоматической классификации транспортных средств в неконтролируемых условиях* // ИТиС, 2015.
- [4] Tianqi Chen, Carlos Guestrin. *XGBoost: A Scalable Tree Boosting System*. <https://arxiv.org/abs/1603.02754>
- [5] Francois Chollet. *Keras* <https://github.com/fchollet/keras> (Github, 2016).
- [6] Tianqi Chen, Tong He, Bing Xu, Michael Benesty, Yuan Tang. *XGBoost*. <https://github.com/dmlc/xgboost> (Github, 2016).
- [7] Fabian Pedregosa, Gael Varoquaux, et al. *Scikit-learn: Machine Learning in Python* // Journal of Machine Learning Research, 12, 2825-2830, 2011.
- [8] E. Burnaev, P. Erofeev. *Influence of Initialization on Learning Time and Accuracy of Nonlinear Regression Model* // Journal of Communications Technology and Electronics, 2016, Vol. 61, No. 6, pp. 646–660.
- [9] E. Burnaev, P. Prikhod'ko. *On a method for constructing ensembles of regression models* // Automation and Remote Control, Volume 74, Issue 10, pp. 1630-1644, 12 Oct 2013.
- [10] Grihon S., Burnaev E.V., Belyaev M.G. and Prikhodko P.V. *Surrogate Modeling of Stability Constraints for*

*Optimization of Composite Structures // Surrogate-Based Modeling and Optimization. Engineering applications. Eds. by S. Koziel, L. Leifsson. Springer, 2013. P. 359-391.*

- [11] M. Belyaev, E. Burnaev. *Approximation of a multidimensional dependency based on linear expansion in a dictionary of parametric functions* // Informatics and Applications, Vol. 7, No. 3, 2013. pp. 114–125.
- [12] Simon S. Haykin. *Neural Networks and Learning Machines*. Prentice Hall, 2009.
- [13] Rafal Jozefowicz, Wojciech Zaremba, Ilya Sutskever. *An Empirical Exploration of Recurrent Network Architectures* // Proceedings of the 32 nd International Conference on Machine Learning, Lille, France, 2015. JMLR: W&CP volume 37.
- [14] Zachary C. Lipton, John Berkowitz, Charles Elkan. *A Critical Review of Recurrent Neural Networks for Sequence Learning*. <https://arxiv.org/abs/1506.00019>
- [15] Bengio, Y., Simard, P., Frasconi, P. *Learning Long term dependencies with gradient descent is difficult* // IEEE Trans Neural Networks. Vol. 5, pp 157–166, 1994.
- [16] Kyunghyun Cho, Bart van Merriënboer, Dzmitry Bahdanau, Yoshua Bengio. *On the properties of neural machine translation: Encoder-decoder approaches*. <https://arxiv.org/abs/1409.1259>
- [17] Theano Development Team. *Theano: A Python framework for fast computation of mathematical expressions*. <https://arxiv.org/abs/1605.02688>
- [18] Martin Abadi, Ashish Agarwal, et al. *TensorFlow: Large-scale machine learning on heterogeneous systems*. <http://arxiv.org/abs/1603.04467>