УДК 004.85 **doi: 10.26906/SUNZ.2018.4.135**

А. В. Нгуен, Я. Е. Сидоров

Национальный аэрокосмический университет имени Н.Е. Жуковского «ХАИ», Харьков, Украина

РЕКУРРЕНТНАЯ НЕЙРОННАЯ СЕТЬ ДЛЯ ОБРАБОТКИ БОЛЬШИХ ТЕКСТОВЫХ ДАННЫХ

Предметом изучения статьи являются нейронные сети, а именно рекуррентные нейронные сети, отличающиеся способностью запоминания данных, а также программные библиотеки для их реализации. **Целью работы** является анализ нейронной сети Хопфилда, сетей Элмана и Джордана, эхо-сети, рекурсивной сети и рекуррентной сети с долгой краткосрочной памятью для непосредственного определения оптимальной архитектуры сети. А также анализ следующих программных библиотек: CNTK, Theano, Gluon, TensorFlow. **Задачи**: провести сравнение по направленности применения и возможностям работы с большими текстовыми данными вышеперечисленных рекуррентных нейронных сетей, определить какая из рассматриваемых программных библиотек является оптимальной и быстродейственной для разработки рекуррентной нейронной сети. **Методом** проведения исследования является нагрузочное тестирование программных комплексов в одинаковых аппаратных условиях, с использованием одинакового набора данных. По итогу работы получены **результаты**: платформой для интеграции технологий выбрано приложение для обработки текстовых данных большого объема и их резюмирования, а именно проект интерактивной среды написания литературы, созданный с использованием .NET, который позволяет автоматически резюмировать текст по определенным критериям. Для анализа производительности программных библиотек был рассмотрен тест на основе обучения и использования рекуррентных сетей с LSTM-модулями на тестовом наборе данных, с использованием всех исследуемых библиотек. **Выводы**: В качестве наиболее оптимального архитектурного подхода стоит считать использование LSTM-модулей, которые решают проблему затухающего градиента. Благодаря этому сети, основанные на этом подходе, показывают наилучшие результаты при работе с долгосрочными зависимостями в данных, что является крайне важным фактором при обработке текстовых данных. По результатам тестов производительности можно сказать, что наиболее оптимизированными для работы с рекуррентными архитектурами являются программные библиотеки CNTK и Gluon. При обучении они демонстрируют скорость, превосходящую производительность TensorFlow и Theano на 10-60%.

Ключевые слов: нейронные сети, рекуррентные сети, программные библиотеки, большие текстовые данные.

Введение

Постановка задачи. На сегодняшний день множество процессов, которые раньше выполняли люди – выполняют различные механизмы, начиная от простых бытовых электрических приборов, заканчивая сложной заводской аппаратурой. Машинное обучение развивается настолько стремительно, что тяжело отследить все направленности, в которых оно актуально уже сегодня. Огромной частью этого направления являются нейронные сети, они настолько распространены, что многие воспринимают машинное обучение и нейронные сети как одно и то же понятие. Это обусловлено эффективностью и доступностью нейронных сетей, а также их возможностями.

Применение для создания новых документов или получения отсортированных данных согласно категориям на основе поданной информации – является огромным преимуществом использования нейронных сетей, поскольку не только упрощает этот процесс, но и уменьшает время, затраченное на поиск. Решение данных задач становится приоритетным для обработки больших текстовых данных.

Целью работы является анализ нейронной сети Хопфилда, сетей Элмана и Джордана, эхо-сети, рекурсивной сети и рекуррентной сети с долгой краткосрочной памятью для непосредственного определения оптимальной архитектуры сети для обработки больших текстовых данных.

1. Рекуррентные нейронные сети

Первая базовая архитектура РНС была разработана еще в 1980-х годах. Она называется полностью рекуррентная сеть. Ее структура заключается в узлах, каждый из которых разделяется на входной, скрытый и выходной. В свою очередь каждый узел сети соединен с каждым другим узлом. Она легла в основу некоторых других реализаций рекуррентных нейронных сетей.

Разнообразие РНС позволяет подобрать необходимую сеть для определенных задач. Некоторые разновидности данной архитектуры узкоспециализированные под конкретные требования и были разработаны специально для них. Для выбора необходимо произвести анализ существующих архитектур, их плюсы и недостатки, возможные сложности реализации, а также сферы, в которых они применяются.

1.1. Нейронная сеть Хопфилда. Нейронная сеть, которая состоит из единственного слоя нейронов. Число нейроном определяется числом входов и выходов. Каждый выход каждого нейрона соединен с входами остальных нейронов по принципу «со всех на все», по этой причине данную нейронную сеть можно назвать полносвязной. Отличительной чертой сети является то, что у нее есть состояния равновесия. Сети Хопфилда работают не до получения ответа, пройдя определенное количество тактов, а до достижения состояния равновесия. Состояние равновесия характерно тем, что следующее состояние сети полностью аналогично предыдущему. Другими словами, можно сказать, что сеть Хопфилда – устойчивая, т.е. может сходиться к единственной фиксированной точке. Подобные точки называются аттракторами, множество этих точек – это память нейронной сети. Для аттрактора существует «область притяжение» - это множество векторов, которые к нему притягиваются. Векторы, кото-

рые попадают в область притяжения становятся связанными с ним. Благодаря чему сеть может действовать как ассоциативная память.

1.2. Сети Элмана и Джордана. По своей структуре сети Элмана и Джордана очень похожи. Сети Джордана характерно иметь дополнительные контекстные нейроны в выходном слое. Эти нейроны на выход принимают информацию от себя и из выходных нейронов, они предназначены для того, чтобы сохранять текущее состояние сети. Также одним из важных требований к сети является то, что контекстных и выходных нейронов должно быть одинаковое количество. В свою очередь, в сети Элмана контекстные нейроны берут вход не от выходных нейронов, а от скрытого слоя нейронов, соответственно число скрытых и контекстных слоев должно быть одинаковым. Это делает ее более гибкой по сравнению с сетью Джордана, поскольку скрытый нейрон намного проще убрать или добавить, чем нейрон выхода. Также сеть Элмана служит основой другой нейронной сети, которая служит для сжатия и шифрования данных. Главной особенностью таких сетей является то, что они хорошо справляются с запоминанием последовательностей, поэтому применяются в системах управления движущимися объектами.

1.3. Эхо-сети. Нейронные эхо сети отличаются своей структурой, поскольку связи между нейронами в них случайны, тогда как в других сетях они организованы аккуратно. Сеть содержит входной слой, скрытый слой, охваченный обратными связями, который также называется «динамическим резервуаром», и выходной слой. Входной слой нейронов служит для инициализации системы, а выходной слой выступает в качестве наблюдателя за порядком активации нейронов относительно времени. В процессе обучения происходит изменение связи между наблюдателем и скрытыми слоями. Также во время обучения обновляются состояния нейронов, и в течение периода времени необходимо следить за выходными данным. Из-за хаотичного расположения связей между нейронами подходы для обучения данного вида сетей ограничены. Использоваться такая сеть может для задач прогнозирования диагностики электроэнергетических систем благодаря своей простоте и скорости обучения.

1.4. Нейронные сети с долгой краткосрочной памятью. Нейронные сети с долгой краткосрочной памятью (LSTM) представляют собой нейронные сети, которые содержат вместо обычных нейронов в скрытых слоях целые вычислительные блоки. Сами блоки содержат критические значения, которые помогают определить следующие параметры: когда необходимо запомнить данные, в какой момент необходимо обратиться к наблюдениям из прошлых итераций и какое значение необходимо отбросить или же забыть, как неудачное. Блоки LSTM сети, которые содержат различные LSTM модули, могут выполнять параллельные вычисления и характерны для «глубоких» многослойных нейронных сетей.

LSTM сеть является универсальной. С соответствующей мастерицей весов, которая рассматривается как программа, и при достаточном числе элементов сети она может выполнять любые вычисления, на которые способен обычный компьютер. Также она обладает относительной невосприимчивостью к длительным временным разрывам, что является существенным преимуществом перед обычными РНС. При обучении и тренировке не происходит размывание и исчезновение данных относительно времени. Данная нейронная сеть является хорошо приспособленной к задачам классификации, даже если важные события растянуты по времени с неопределённой продолжительностью и разрывами. Более того, LSTM-сети достигли наилучшего результата в распознавании рукописного текста. Прекрасные результаты LSTM-сеть показала и в следующих задачах: анализ временных рядов, распознавание человеческой активности, грамматическое обучение, ритмическое обучение, генерация музыкальных произведений, робототехника.

1.5. Рекурсивная сеть. Рекурсивные нейронные сети работают с данными переменной длины. Подобные сети могут состоять из нескольких модулей, выходной сигнал одного модуля подается на вход другого модуля того же типа. Результаты нескольких модулей объединяются в один – ассоциирующий модуль. Вследствие такой архитектуры данные сети также называют сетями древовидной структуры. Лучше всего они справляются с задачами распознавания естественного языка, например, для определения тональности предложения. В процессе работы фразы и предложения моделируются через векторное представление слов.

2. Библиотеки для реализации нейронных сетей

На сегодняшний день существует ряд готовых решений, созданных большими корпорациями и академическими группами. Эти решения существенно упрощают процесс разработки и обучения искусственного интеллекта, предоставляя различный уровень абстракций. Можно выделить низкоуровневые фреймворки, которые реализуют математическую логику, позволяя пользователю самому реализовывать логические единицы нейронной сети. Также существуют фреймворки более высокого уровня, которые предоставляют готовые абстракции для большего удобства и скорости разработки системы.

2.1. TensorFlow – программная библиотека для глубокого обучения, разработанная Google для решения задач построения и тренировки нейронных сетей. На данный момент является одной из самых широко применяемых библиотек как для научных исследований, так и для коммерческих проектов. Является продолжением закрытого проекта DistBelief, который создавался Google для внутреннего использования. Данная библиотека реализована на Python, также имеются официальные реализации для C++, Haskell, Java и Go. Однако для других популярных языков программирования имеются неофициальные обертки, разработанные пользователями. TensorFlow основана на графах операций, которые оперируют с тензорными вычислениями. Таким образом библиотека представляет собой низкоуровневый инструментарий без привязки к конкрет-

ным объектам нейронной сети. Производительность данной библиотеки заключается в использовании символьного подхода к вычислениям.

2.2. Theano – библиотека, которая используется для разработки систем машинного обучения как сама по себе, так и в качестве вычислительного бекэнда для более высокоуровневых библиотек, например, Lasagne, Keras или Blocks. Theano разрабатывается с 2007 года, главным образом, группой MILA из Университета Монреаля и названа в честь древнегреческой женщины-философа и математика Феано. Основными принципами являются: интеграция с numpy, прозрачное использование различных вычислительных устройств (обычно GPU), динамическая генерация оптимизированного C-кода. Theano, как и TensorFlow, использует символьный подход к вычислениям.

2.3. Gluon – программная библиотека для реализации нейронных сетей, разработанная совместно компаниями Microsoft и Amazon. Ее характерными особенностями являются быстрое прототипирование, оптимизация для работы в облачных службах, распараллеливание вычислений и упор на оптимизацию LSTM-модулей и рекуррентных-сетей. Также Gluon поддерживает работу с разряженными и квантованными данными, что является большим преимуществом при работе с естественным языком. Gluon ориентирован для работы с облачными система и обладает для этого широким API. В свою очередь это позволяет поддерживать сложные методы, например, динамические графы и гибкие структуры, без необходимости разбираться в конкретных деталях и оптимизировать все вручную.

2.4. Cognitive Toolkit (CNTK) – бесплатная программная библиотека с открытым исходным кодом, разработанная для глубокого машинного обучения компанией Microsoft. На сегодняшний день является одной из самых популярных библиотек для построения нейронных сетей и главным конкурентом вышеописанного TensorFlow. Из основных особенностей и отличий можно выделить:

- скорость. CNTK в целом работает быстрее, чем TensorFlow, а в рекуррентных сетях дает вплоть до пяти- и десятикратного выигрыша в производительности;
- структура API. CNTK имеет гибкий и мощный API для C++ и предлагает как низкоуровневые, так и простые в использовании высокоуровневые Python API на основе парадигмы функционального программирования;
- масштабируемость. CNTK легко масштабируется и в случае вычислительно требовательных задач может выполняться хоть на тысячах графических процессоров;
- скоринг. В CNTK есть производительный Eval API для C++, .NET, Java и Python, для упрощения интеграции нейронных сетей в свои приложения;
- расширяемость. CNTK легко расширяется благодаря возможности использования Python для определения собственных слоев и процедур обучения.
- встроенные модули считывания. В CNTK есть нетребовательные к памяти встроенные средства чтения данных.

3. Реализация обработки данных

Целевой платформой для интеграции рассмотренных в данной статье технологий может быть приложение для обработки текстовых данных большого объема и их резюмирования. Таковым является проект интерактивной среды написания литературы, созданный с использованием .NET, который позволяет автоматически резюмировать текст по определенным критериям. Ключевую роль в выполнении подобной задачи играет быстродействие и низкое потребление ресурсов компьютера или планшета, что подразумевает под собой оптимизацию на уровне программной библиотеки машинного обучения. В качестве наиболее оптимального архитектурного подхода стоит считать использование LSTM-модулей, которые решают проблему затухающего градиента. Благодаря этому сети, основанные на этом подходе, показывают наилучшие результаты при работе с долгосрочными зависимостями в данных, что является крайне важным фактором при обработке текстовых данных. Для анализа производительности программных библиотек был рассмотрен тест на основе обучения и использования рекуррентных сетей с LSTM-модулями на тестовом наборе данных, с использованием всех вышеупомянутых библиотек (рис. 1). Тестирование производительности производилось на аппаратной платформе Amazon EC2 с вычислительным CUDA-процессором Nvidia Tesla K80.

Рис. 1. Анализ производительности на основе LSTM

По результатам тестов производительности можно сделать вывод, что наиболее оптимизированными для работы с рекуррентными архитектурами являются программные библиотеки CNTK и Gluon. При обучении они демонстрируют скорость, превосходящую производительность TensorFlow и Theano на 10-60%, что можно увидеть из графиков на рис. 1 и 2.

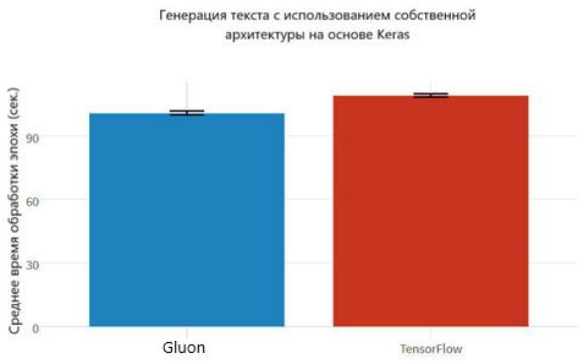

Рис. 2. Анализ производительности на основе собственной архитектуры

CNTK поставляется с функцией использования его в качестве вычислительной основы для высокоуровневой библиотеки Keras, что делает ее предпочтительной для использования в задачах обработки натурального языка и работе с большими данными. Рассматривается вариант написания собственной программной платформы для машинного обучения с использованием .NET. С учетом того, что такая реализация предоставит максимальный уровень интеграции с целевой системой, это позволит избежать потерь в производительности.

Выводы

По итогам исследования можно выделить одну из разновидностей рекуррентных нейронных сетей, которая больше всего подходит для работы с текстом. Это нейронная сеть с долгой краткосрочной памятью, поскольку она показала максимальные результаты в данной области. Свойство LSTM-сети запоминать информацию во времени с неопределенной продолжительностью делает ее наиболее эффективным инструментом для обработки долгосрочных зависимостей при работе с натуральным языком. Рекурсивная нейронная сеть также ориентирована на работу с текстом и распознавание естественного языка, однако, она больше направлена на работу с тональностями в пределах предложений и небольших абзацев. В рамках задачи работы с большими текстовыми данными подобное ограничение является неприемлемым, поскольку приведет к потере необходимых данных и невозможности установить связи между характеристиками, заданными через большие интервалы текста. В качестве программной основы для реализации рекуррентных нейронных сетей, наиболее эффективным подходом будет использование CNTK и Gluon, которые оптимизированы для работы с LSTM-модулями.

СПИСОК ЛИТЕРАТУРЫ

1. Levy Omer, Yoav Goldberg, Ido Dagan. Improving distributional similarity with lessons learned from word embeddings. Transactions of the Association for Computational Linguistics 3. — 2015. — P. 211–225.
2. Дональд Мичи. Machine Learning, Neural and Statistical Classification. 1994г. Лос-Аламос. с.120-145
3. Christopher Manning. Computational linguistics and deep learning. Computational Linguistics. — 2016.
4. Кристофер Бишоп. Pattern Recognition and Machine Learning. 2006г. Массачусетс. с.201-215
5. Кевин Мерфи. Machine Learning: A Probabilistic Perspective. 2012г. Технический колледж Нью-Йорка. с.65-82

Рецензент: д-р техн. наук, проф. К. С. Козелкова,
Дкржавний університет телекомунікацій, Київ
Received (Надійшла) 28.06.2018
Accepted for publication (Прийнята до друку) 29.08.2018

Рекурентна нейронна мережа для обробки великих текстових даних

А. В. Нгуєн, Я. Є. Сідоров

Предметом вивчення статті є нейронні мережі, а саме рекурентні нейронні мережі, що відрізняються здатністю запам'ятовування данні, а також програмні бібліотеки для їх реалізації. **Метою** роботи є аналіз нейронної мережі Хопфілда, мереж Елмана і Джордана, ESN-мережі, рекурсивної мережі і рекурентної мережі з довгою короткостроковою пам'яттю для безпосереднього визначення оптимальної архітектури мережі. А також аналіз наступних програмних бібліотек: CNTK, Theano, Gluon, TensorFlow. **Завдання**: провести порівняння за спрямованістю застосування і можливостям роботи з великими текстовими даними перерахованих вище рекурентних нейронних мереж, визначити яка з розглянутих програмних бібліотек є оптимальною і швидкодійною для розробки рекурентної нейронної мережі. **Методом** проведення дослідження є тестування навантаження програмних комплексів в однакових апаратних умовах, з використанням однакового набору даних. За підсумком роботи отримано **результати**: платформою для інтеграції технологій вибрано додаток для обробки текстових даних великого обсягу і їх резюмування, а саме проект інтерактивного середовища написання літератури, створений з використанням .NET, який дозволяє автоматично резюмувати текст за певними критеріями. Для аналізу продуктивності програмних бібліотек було розглянуто тест на основі навчання та використання рекурентних мереж з LSTM-модулями на тестовому наборі даних, з використанням усіх досліджуваних бібліотек. **Висновки**: в якості найбільш оптимального архітектурного підходу варто вважати використання LSTM-модулів, які вирішують проблему затухаючого градієнта. Завдяки цьому мережі, засновані на цьому підході, показують найкращі результати при роботі з довгостроковими залежностями, що є вкрай важливим фактором при обробці текстових даних. За результатами тестів продуктивності можна сказати, що найбільш оптимізованими для роботи з рекурентною архітектурою є програмні бібліотеки CNTK і Gluon. При навчанні вони демонструють швидкість, що перевершує продуктивність TensorFlow і Theano на 10-60%.

Ключові слова: нейронні мережі, рекурентні мережі, програмні бібліотеки, великі текстові дані.

Recurrent neural network for processing large text data

A. Nguyen, Y. Sidorov

The **subject** of the article is neural networks, namely recurrent neural networks, characterized by the ability to store long-term data dependencies, as well as software libraries for machine learning implementation. The **aim** of the work is to analyze the Hopfield neural network, Elman and Jordan networks, echo network, recursive network and recurrent network with long short-term memory to directly determine the most efficient network architecture. And also this paper contains analysis of the following software libraries: CNTK, Theano, Gluon, TensorFlow. **Tasks**: to compare the software tools pros and cons of application and the possibilities of working with large text data using neural networks described above, to determine which of the software libraries under consideration is optimal and performance-effective for the development of a recurrent neural network. The method of the study is load testing of software frameworks in the same hardware conditions, using the same set of data. The **results** of the work are following: the technology integration platform is picked to be an application for large-scale text data processing and its summarization, namely the project of an interactive writing environment created using .NET, which allows to automatically summarize the text according to certain criteria. To analyze the performance of software libraries, an examination of a premade benchmark-test has took place. A benchmark test, at its core, is based on the training and use of recurrent networks with LSTM-modules on a test set of data, using all of the considered frameworks. **Conclusion**: As the most optimal architectural approach, it is worth considering using an LSTM-modules, which solves the problem of a vanishing gradient. Thanks to this, neural networks based on this approach show the best results when working with long-term dependencies in data, which is an extremely important factor in a text data processing. According to the results of performance tests, we might consider using CNTK and Gluon as they appear to be the most optimized solution for LSTM-units applying. When training, they demonstrate a speed that exceeds the TensorFlow and Theano performance by 10-60%.

Keywords: neural networks, recurrent networks, program libraries, large text data.