2 СИСТЕМНОЕ ПРОЕКТИРОВАНИЕ

После изучения теоретических аспектов разрабатываемой системы и выработки списка требования для разработки, требуется выделить отдельные функциональные блоки. Разделение системы на более мелкие составные части позволяет упростить процесс разработки и в итоге получить более гибкую систему.

В структуре разрабатываемого проекта можно выделить следующие модули (см. схему ГУИР.400201.139 С1):

- библиотека шаблонов;
- библиотека компонентов;
- модуль проектирования сети;
- модуль ядра;
- модуль обучения;
- модуль визуализации;
- модуль оптимизации;
- модуль тестирования;
- модуль интеграции с инструментами параллельных вычислений.

Центральными модулями системы являются ядро, библиотека компонентов и библиотека шаблонов. С их помощью создается структура нейронной сети, происходит преобразование параметров сети во время работы системы. Пользователь не взаимодействует с этими блоками напрямую т.к. функционал низкого уровня скрыт за внешним интерфейсом. С помощью модуля обучения реализуется специальный режим работы сети в котором ей подаются обучающие наборы входных данных, где для каждой совокупности входных данных уже определено верное значение, которое должна выдать нейронная сеть на выходе. После определенного количества циклов обучения, система принимает свойство выдавать корректные (с определенным процентом точности) результаты для любых заданных конфигураций на входе. Так как работа даже сравнительно небольшой нейронной сети сопряжена с обработкой больших объемов численных данных, необходимо позаботиться об оптимизации вычислений. Модуль интеграции с инструментами параллельных вычислений позволяет при возможности распараллеливать обработку данных, используя многопоточность. Кроме этого, известно, что во время работы нейронных сетей большинство вычислительных операций сводится к перемножению матриц и векторов. Поэтому большой выигрыш в производительности может дать использование возможностей процессора по векторизации. Разумеется, использование многопоточности на сегодняшний день доступно не всегда, поэтому система имеет возможность работы в рамках одного потока (с определенным снижением производительности).

При разработке нейронных сетей большое значение имеет выбор основных параметров работы сети. Часто возникает такая ситуация, что разработанная система машинного обучения работает неэффективно, но

конкретная причина ошибок не всегда известна. Вариантов решения возникших проблем может быть множество и далеко не всегда очевидно какой из них стоит применить. Нейронная сеть может показывать отличные результаты на обучающем наборе, но при этом работать неэффективно с новыми данными. Некоторые методы решения проблем некорректной работы сети требуют больших материальных и временных вложений (например, сбор большого количества обучающих данных, или определение нового вектора признаков). Из-за этого большую ценность для разработчика системы машинного обучения представляют данные, которые позволяют лучше понять, в чем заключаются проблемы в работе и как найти методы их решения. Такие данные включают в себя различные оценки работы сети (точность предсказания), а также графики (кривая обучения, график изменения функции стоимости).

Данный функционал реализован в блоках визуализации и тестирования. С их помощью пользователь может проводить диагностику системы, выявлять возможные ошибки на этапе разработки, получать отчеты об эффективности работы сети.

Рассмотрим каждый модуль системы более подробно.

Модуль ядра содержит в себе основную логику работы нейронной сети. Именно здесь задаются принципы работы всех типов слоев, доступных клиенту. В данном блоке происходят все математические вычисления, связанные с функционированием нейронной сети. Модуль использует библиотеку Apache Commons Math, которая дает доступ к множеству математических абстракций и операций над ними, в том числе:

- дифференцирование функций
- минимизация функций
- работа с матрицами и векторами;
- статистические функции;
- решение систем линейных уравнений.

Данная библиотека распространяется под свободной лицензией, обладает обширной документацией и широко используется разработчиками Java в большом числе проектов.

Модуль ядра принимает запросы от остальных модулей системы и в соответствии с этим изменяет текущее состояние нейронной сети. Например, во время процесса обучения сети модуль получает массив обучающих наборов, которые необходимо пропустить через сеть, вычислить значение функции стоимости, рассчитать новые значения для коэффициентов сети и преобразовать их. Также модуль ядра ответственен за выдачу данных о текущем состоянии сети модулю визуализации.

Библиотека компонентов состоит из базовых абстракций, которые использует модуль ядра для построения архитектуры нейронной сети. Библиотека содержит в себе описания таких понятий как сеть, слой сети, вектор признаков, обучающий набор. Было принято решение не реализовывать отдельные модели для нейронов сети и связей между ними.

Базовым блоком для конструирования сети является слой. Это позволяет упростить общую архитектуру системы, снизить затраты на память и улучшить время работы. Деление каждого слоя на отдельные нейроны имеет смысл только для создания большей наглядности при изучении принципов машинного обучения.

Большое число различных типов слоев для конструирования нейронной сети позволяет создавать системы, пригодные для решения широкого класса задач. Перечислим некоторые из моделей слоев, предоставляемых библиотекой:

1. Сверточный слой — это основной блок сверточной нейронной сети. Слой включает в себя фильтр — ядро свёртки, который обрабатывает предыдущий слой по фрагментам (суммируя результаты матричного произведения для каждого фрагмента). Весовые коэффициенты ядра свёртки (небольшой матрицы) неизвестны и устанавливаются в процессе обучения.
2. Слой ReLU — сокращение от английского Rectified Linear Unit и означает блок линейной ректификации. Слой ReLU — ни что иное как функция активации после сверточного слоя, однако для активации выбирается вместо обычных функций ненасыщаемая функция сравнения с нулем. Такая функция показывает хорошие результаты при обучении нейронных сетей и отвечает за отсечение ненужных деталей в канале (при отрицательном выходе).
3. Слой пулинга (иначе подвыборки, субдискретизации) представляет собой нелинейное уплотнение карты признаков, при этом группа пикселей (обычно размера 2×2) уплотняется до одного пикселя, проходя нелинейное преобразование. Наиболее употребительна при этом функция максимума. Преобразования затрагивают непересекающиеся прямоугольники или квадраты, каждый из которых ужимается в один пиксель, при этом выбирается пиксель, имеющий максимальное значение.
4. Softmax слой, который позволяет нормализовать значения выходного слоя сети. Все что делает этот слой – делит значение каждого нейрона на входе на сумму значений всех входов. Такое преобразование обеспечивает сумму выходов слоя, равную единице при любых значениях сигнала данного слоя. Это позволяет трактовать выход как вероятность событий, совокупность которых образует полную группу.
5. Слои реализующие различные функции активации (сигмоиду, гипертангенс). В сверточной нейронной сети такие слои обычно разделяют слои свертки и пулинга. Вся их работа заключается в применении определенной функции к каждому входу слоя.

В любом случае наличие модуля библиотеки оставляет возможности для будущего расширения функциональности путем добавления новых элементов.

Модуль обучения предоставляет пользователю интерфейс для обучения нейронной сети. В модуле реализованы различные алгоритмы обучения, с помощью которых нейронная сеть приобретает способность к самостоятельному функционированию. Пользователь должен обеспечить

модуль массивом обучающих данных, которые будут переданы ядру сети для обработки. Данные могут быть переданы в виде текстовых файлов либо как результаты работы другой части системы. В случае импорта обучающих данных в виде текстовых файлов модуль обучения должен проверить корректность этих данных. После того как обучающие наборы были загружены модуль начинает выполнять один из алгоритмов обучения. При этом модуль обучения контролирует работу модуля ядра.

Модуль интеграции с инструментами параллельной обработки использует методы, предоставляемые языком Java для распараллеливания вычислений. Как было сказано ранее, во время работы нейронной сети большую часть времени занимают операции умножения матриц и векторов. Эти операции хорошо поддаются распараллеливанию и векторизации. При корректном использовании многопоточности можно получить выигрыш по производительности в 3-4 раза.

Пакет `java.util.concurrent` предоставляет множество инструментов для эффективного использования многопоточности, в том числе:

1. Многопоточные коллекции — набор коллекций, более эффективно работающие в многопоточной среде нежели стандартные универсальные коллекции языка Java. Вместо базового класса-обертки с блокированием доступа ко всей коллекции используются блокировки по сегментам данных или же оптимизируется работа для параллельного чтения данных по wait-free алгоритмам.
2. Очереди — неблокирующие и блокирующие очереди с поддержкой многопоточности. Неблокирующие очереди заточены на скорость и работу без блокирования потоков. Блокирующие очереди используются, когда нужно «притормозить» потоки производителя или потребителя, если не выполнены какие-либо условия, например, очередь пуста или переполнена, или же нет свободного потребителя.
3. Исполнители — содержит в себе отличные фрейморки для создания пулов потоков, планирования работы асинхронных задач с получением результатов.

Модуль визуализации представляет собой веб-приложение, созданное при помощи фреймфорка Spring. Приложение получает данные во время работы сети от модуля ядра, обрабатывает эти данные и представляет результат на веб-странице. Было принято решение использовать в качестве модуля визуализации именно веб-приложение, так как на сегодняшний день существует огромное количество различных плагинов и библиотек, позволяющих визуализировать данных любого рода. Инструменты для создания десктопных приложений не могут обеспечить такой широкий спектр возможностей. В это же время использование веб-приложения обеспечивает большую гибкость и платформонезависимость.

Так как модуль ядра и связанные с ним библиотеки никак не зависят от модуля визуализации (с точки зрения паттерна MVC вычислительное ядро сети является моделью), то существует возможность создания новых модулей,

предназначенных для визуализации данных. Так, язык Java является основным языком для разработки приложений на платформе Android, поэтому логичной представляется разработка мобильного приложения на основе существующей библиотеки. Также возможна разработка десктопной версии, если это необходимо.

Модуль проектирования сети предоставляет клиенту интерфейс для создания конфигурации нейронной сети. При помощи данного блока пользователь приложения создает требуемую архитектуру сети, задает параметры отдельных слоев и всей системы в целом. В задачи этого блока входит проверка корректности данных, введенных пользователем и выработка оповещений при возникновении ошибок (например, при некорректно заданных размерностях слоев). После проверки введенных данных происходит передача структуры сети в модуль ядра где происходит инициализация сети.

Модуль тестирования предназначен для проверки работы нейронной сети после процесса обучения. В данном режиме работы сеть пытается предсказать результат имея только определенный набор входных данных. Коэффициенты нейронов при этом не изменяются. Таким образом пользователь может обнаружить возможные ошибки проектирования, узнать насколько эффективно была обучена сеть и требуется ли дополнительное обучение.