1 ОБЗОР ЛИТЕРАТУРЫ

1.1 Обзор существующих аналогов

Благодаря тому, что машинное обучение стремительно развивалось последние несколько лет, сегодня существует большое количество различных фреймворков и библиотек для решения задач в этой области. Проекты, позволяющие интегрировать алгоритмы машинного обучения с собственными системами можно найти практически для всех известных платформ. Существующие аналоги были изучены на этапе проектирования. Рассмотрим некоторые из них.

TensorFlow – библиотека машинного обучения, изначально разработана специалистами научно-исследовательского подразделения Google для внутренних нужд компании, но в ноябре 2015 года была выпущена под свободной лицензией [1]. В состав библиотеки входит модуль TensorBoard реализованный в виде веб-приложения и предназначенный для визуализации работы системы и выдачи статистической информации (см. рисунок 1.1).

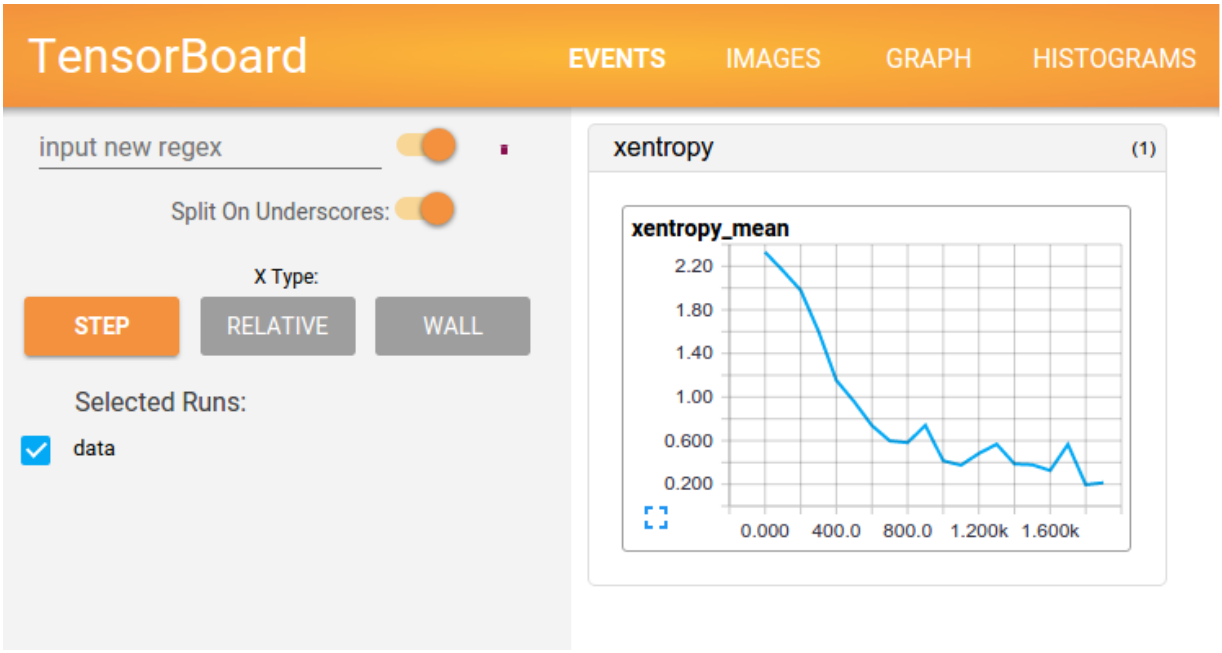

Рисунок 1.1 – Окно модуля TensorBoard

Особенностью библиотеки является ее модель исполнения. Центральным объектом TensorFlow является граф потока данных, представляющий вычисления. Вершины графа представляют операции, а ребра – тензоры (многомерные массивы, являющиеся основой TensorFlow). Граф потока данных в целом является полным описанием вычислений, которые реализуются в рамках сессии и выполняются на устройствах вычисления (см. рисунок 1.2). Узлы графа могут закрепляются за вычислительными устройствами и выполняться асинхронно, параллельно обрабатывая разом все подходящие к ним тензоры. Таким образом строится

нейронная сеть, все узлы которой работают одновременно по аналогии с одновременной активацией нейронов в мозге.

Преимуществом библиотеки TensorFlow является ее масштабируемость в совокупности с широким спектром задач, при решении которых она может быть применена. Существует API для языков Python, C++, Java и Go. Библиотека может использовать ресурсы CPU и GPU, при использовании графических процессорах видеокарт возможны расчёты в приложениях для общих вычислений с помощью дополнительного расширения CUDA. TensorFlow работает на 64-битных серверах и настольных компьютерах Linux, Windows и Mac OS X, а также на мобильных платформах, в том числе Android и iOS. Компания Google, а с недавнего времени и многие другие крупные организации используют TensorFlow практически повсеместно, от распознавания речи до поиска фотографий.

Кроме этого, к преимуществам можно отнести и обширную документацию, что облегчает изучение возможностей библиотеки и снижает порог вхождения. После открытия исходного кода библиотеки, репозиторий проекта на сайте GitHub стал одним из лидеров по числу ответвлений и коммитов. Это говорит о том, что в развитии проекта заинтересовано большое количество специалистов и в ближайшее время следует ожидать исправления недостатков, повышения производительности и реализации новых функций.

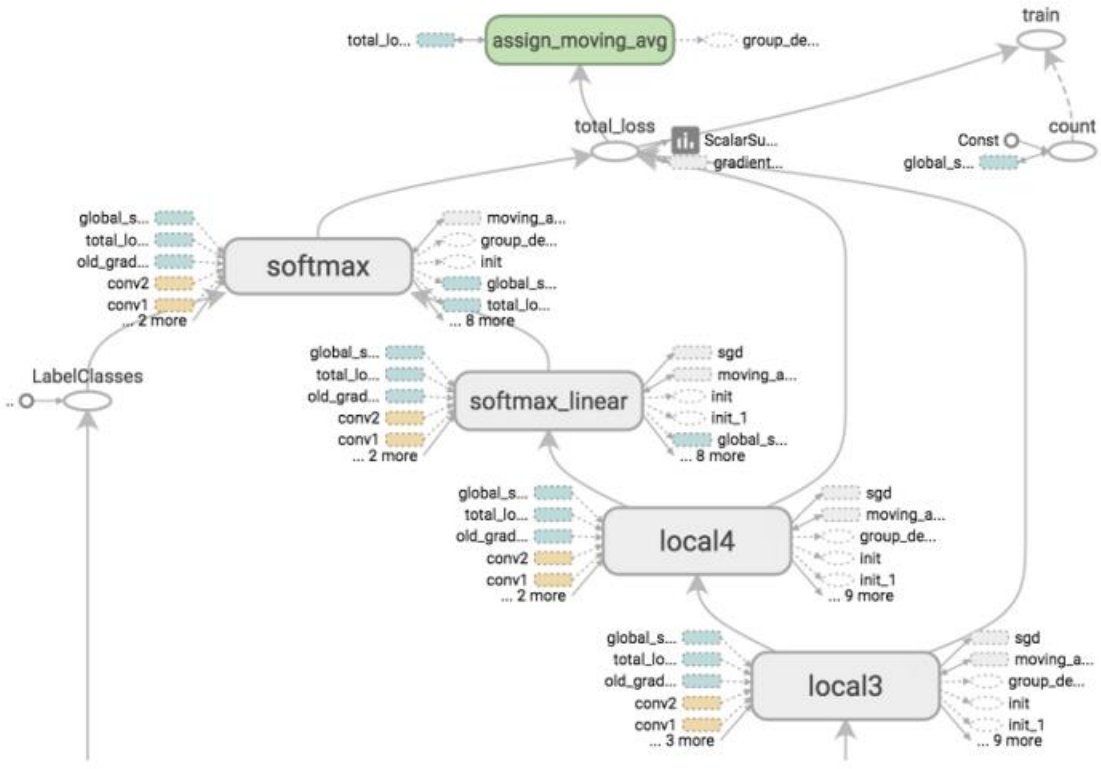

Рисунок 1.2 – Пример визуализации графа вычислений с помощью TensorBoard

К недостаткам данной библиотеки можно отнести тот факт, что она не ориентирована исключительно на нейронные сети, в связи с чем проигрывает

по производительности другим библиотекам, оптимизированным для работы с нейронными сетями [2]

Также, как было сказано выше, библиотека была представлена сравнительно недавно и находится на этапе активной разработки. В связи с этим многие заявленные функции библиотеки реализованы не в полном объеме. Как показывает практика, ранние версии таких крупных проектов как TensorFlow могут содержать ошибки. Имеет смысл дождаться выхода новых, более стабильных версий библиотеки

Еще одним примером системы, предназначенной для разработки нейронных сетей можно назвать MATLAB - пакет прикладных программ для решения задач технических вычислений, популярный среди инженерных и научных работников. Пакет был разработан в конце 1970-х годов Кливом Моулером, занимающим должность декана факультета компьютерных наук в Университете Нью-Мексико, с целью разработки было избавить студентов факультета от необходимости изучения языка программирования Фортран и создать среду, пригодную для научных и инженерных вычислений [3].

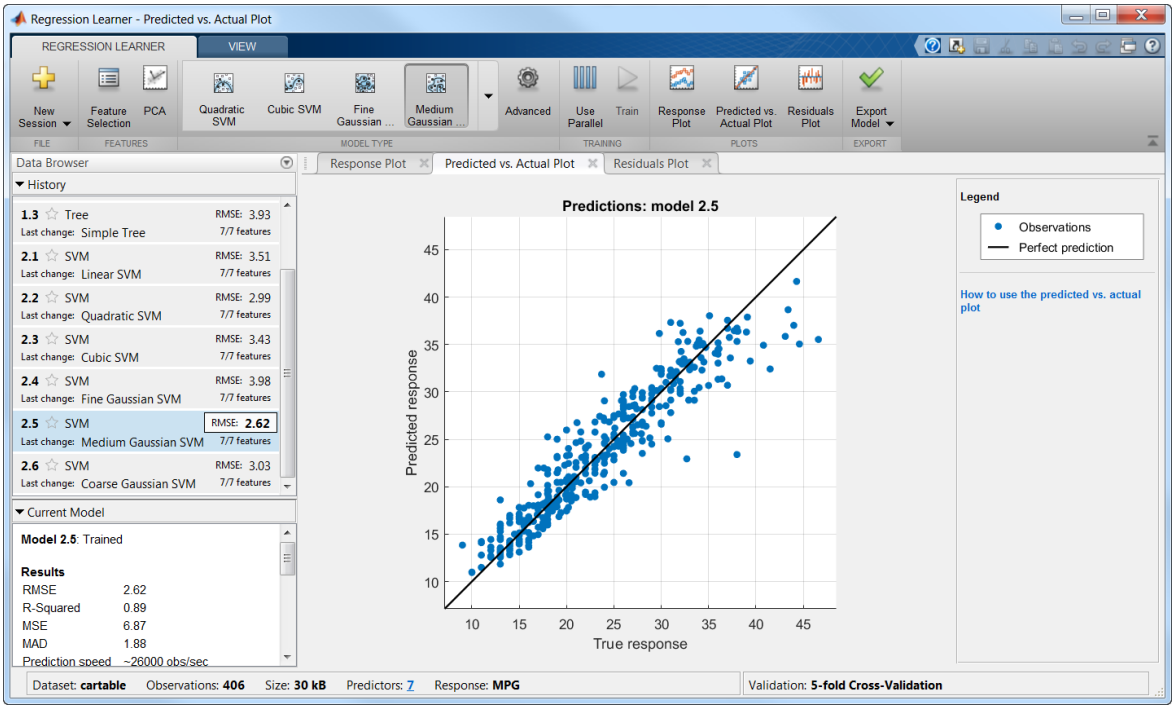

Рисунок 1.3 – Визуализация решения задачи линейной регрессии в среде MATLAB

Пакет MATLAB включает в себя одноименный язык программирования, предоставляет средства для разработки алгоритмов включая высокоуровневые с использованием концепций объектно-ориентированного программирования. Кроме этого в составе пакета имеются функции для построения графиков, в том числе трёхмерных, визуального анализа данных и создания анимированных роликов (см. рисунок 1.3). Встроенная среда разработки позволяет создавать графические интерфейсы пользователя с различными элементами управления, такими как кнопки и поля ввода.

Однако, несмотря на все вышеперечисленные особенности, пакет имеет ряд недостатков, которые не позволяют эффективно использовать его при решении задач машинного обучения.

Во-первых, пакет не содержит в себе достаточное количество функциональных элементов для создания нейронных сетей. Так как инструмент изначально не был разработан для работы с машинным обучением, то все возможности по работе с нейронными сетями реализованы в виде дополнительных расширений.

Одним из основных минусов пакета MATLAB является крайне неудобный интерфейс. Большое количество окон и элементов управления могут серьезно затруднить разработку и тестирование систем. Множество функций, доступных разработчику и как следствие большой объем документации затрудняют изучение в том случае, если специалист не обладает опытом работы с пакетом. Плюс к этому, в пакете используется собственный язык программирования, что также усложняет разработку. Также это создает проблемы в том случае, когда необходима интеграция с другими частями системы, реализованными на языках C#, C++, Java. Для того чтобы начать работать с MATLAB необходимо либо потратить время на изучение нового языка, либо нанять специалистов, имеющих опыт работы с данным инструментом. И то, и другое только повышает стоимость разработки приложения.

Еще одним недостатком MATLAB является серьезное снижение производительности при работе с данными большого объема. При разработке реальных проектов с использованием нейронных сетей размеры массивов данных, используемых в качестве обучающих наборов могут достигать больших значений (от десятков миллионов до миллиардов элементов). Качественная система разработки должна быть оптимизирована для работы с таким количеством информации.

Следует упомянуть о том, что инструментарий распространяется под коммерческой лицензией. Стоимость лицензии MATLAB для коммерческих организаций составляет около $2880, библиотеки визуального моделирования Simulink - $4220, необходимые библиотеки докупаются отдельно (к примеру, стоимость библиотеки Real-Time WindowsTarget для модуля Simulink составляет $3000). Дороговизна лицензии может стать серьезным препятствием для использования пакета [4].

1.2 Глубинные нейронные сети

Глубинное обучение — это класс алгоритмов машинного обучения, который использует многослойную систему нелинейных фильтров для извлечения признаков с преобразованиями. Каждый последующий слой при этом получает на входе выходные данные предыдущего слоя. По определению любая нейронная сеть с более, чем одним скрытым слоем, считается глубинной [5].

Система глубинного обучения может сочетать алгоритмы обучения с учителем и без учителя, при этом анализ образца представляет собой обучение без учителя, а классификация - обучение с учителем.

Данный класс обучения обладает несколькими слоями выявления признаков или параметров представления данных. формирует в процессе обучения слои на несколько уровнях представлений, которые соответствуют различным уровням абстракции; слои образуют иерархию понятий.

Эти признаки организованы иерархически, признаки более высокого уровня являются производными от признаков более низкого уровня [6].

Сегодня глубинное обучение используется в совершенно разных сферах, но, пожалуй, больше всего примеров использования лежит в области обработки изображений. Некоторые представления позволяют легче решать поставленные задачи, например, распознавание лица или распознавание выражения лица [7]. Одна из основных проблем при распознавании изображений – определение набора признаков, по которым следует производить классификацию. Изображение представляется компьютеру как множество чисел и признаки, которые замечает человек не могут быть обнаружены машиной.

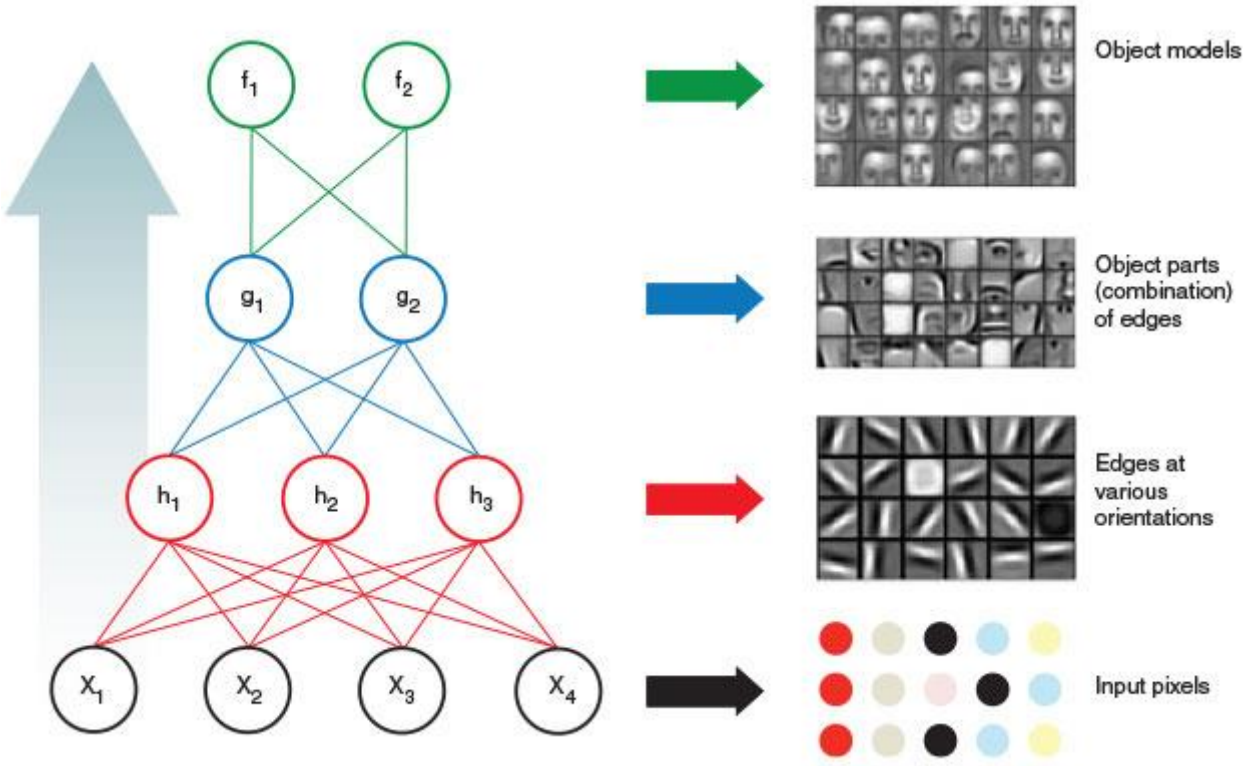

Рисунок 1.4 – Пример работы алгоритма глубинного обучения

Многослойная структура алгоритма позволяет постепенно переходить от распознавания простейших структур к все более и более абстрактным формам. Так, в случае алгоритма распознавания изображений, первые слои ответственны за поиск различных линий, отрезков, изгибов, углов и т.п. Основываясь на результатах работы первых слоев, на последующих шагах происходит идентификация все более сложных признаков (см. рисунок 1.4). В

итоге вместо того, чтобы пытаться одновременно анализировать весь массив пикселей исходного изображения, мы получаем иерархическую структуру признаков, которая позволяет решать гораздо более сложные задачи с меньшими затратами времени и машинных ресурсов [8].

1.3 Фреймворк Spring

Spring – универсальный фреймворк для создания приложений на платформе Java, предоставляющий набор расширений, которые помогают значительно ускорить процесс разработки и тестирования. Первая версия фреймворка была представлена в июне 2003 года. Последняя на текущий момент версия – 4.0, была выпущена в 2013 году. Spring пользуется большой популярностью у разработчиков, по позволяет существенно упростить разработку любых Java приложений на всех этапах, начиная от организации взаимодействия с базами данных и заканчивая модульным тестированием приложения. Основные принципы работы с Spring описаны в [9].

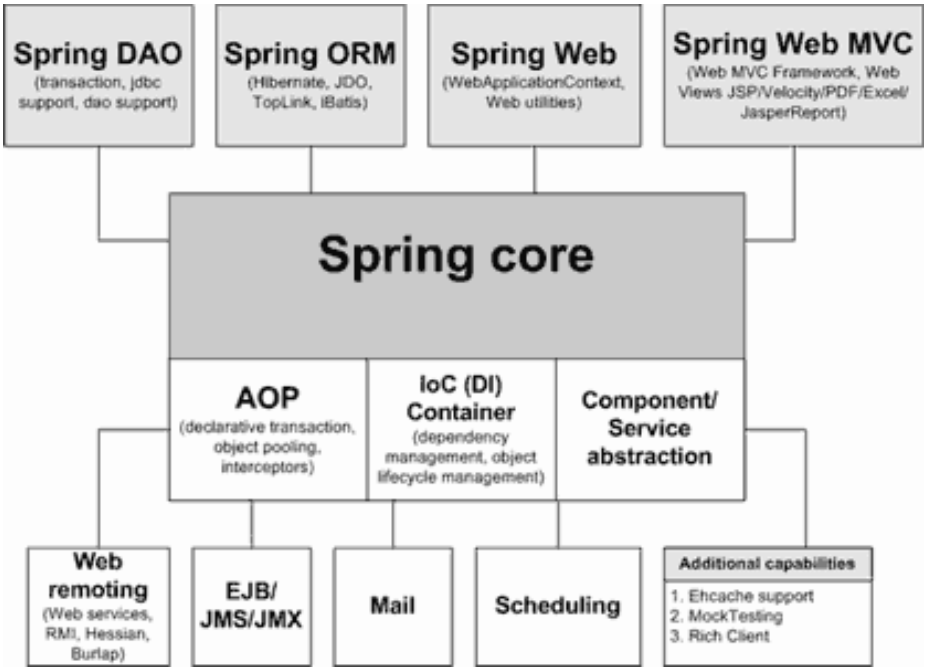

Рисунок 1.5 – Основные модули фреймворка Spring

В состав фреймворка входят следующие модули (см. рисунок 1.5):

- Inversion of Control контейнер;
- фреймворк аспектно-ориентированного программирования;
- фреймворк доступа к данным;
- фреймворк управления транзакциями;
- фреймворк аутентификации и авторизации;
- веб-фреймворк;
- модуль тестирования.

Каждый из модулей может быть использован независимо от других – это позволяет подключить лишь тот функционал, который необходим для конкретного приложения. Многие расширения, которые предоставляет Spring (аспектно-ориентированное программирование, управление транзакциями) невозможно, либо крайне трудно реализовать самостоятельно, что делает данный фреймворк незаменимым при разработке приложений любого рода.

Так как на сегодняшний день Spring используется практически повсеместно, огромное сообщество специалистов, использующих фреймворк, позволяет быстро решить любые проблемы, возникающие во время разработки. Возможности фреймворка позволяют интегрировать в приложение модули, созданные без использования Spring.

1.4 Шаблон проектирования MVC

Шаблон MVC (Model-View-Controller) разделяет работу приложения на три отдельные функциональные роли: модель данных (Model), представление (View) и управляющую логику (Controller). Таким образом, изменения, вносимые в один из компонентов, оказывают минимально возможное воздействие на другие компоненты (см. рисунок 1.6).

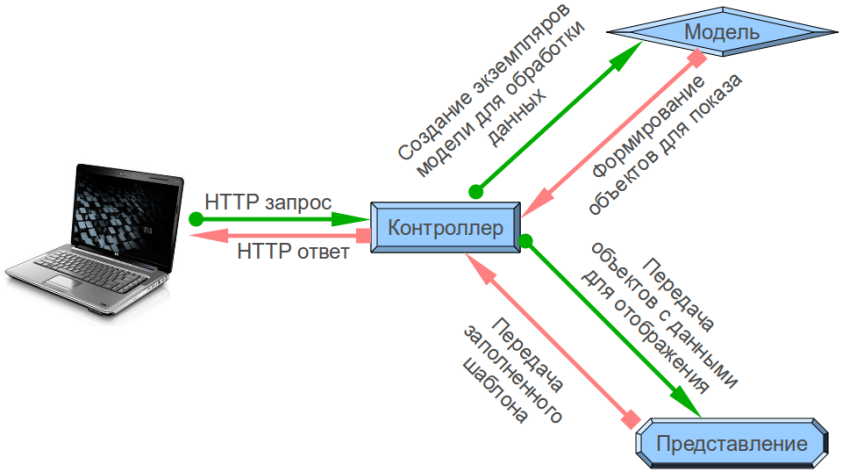

Рисунок 1.6 – Диаграмма паттерна MVC

Впервые этот шаблон был применён в фреймворке, разрабатываемом для языка Smalltalk в конце 1970-х годов. Разработчики должны были придумать архитектурное решение, которое позволяло бы отделить графический интерфейс от бизнес логики, а бизнес логику от данных. Таким образом, в классическом варианте, MVC состоит из трех частей, которые и дали ему название. Под моделью, обычно понимается часть, содержащая в себе функциональную бизнес-логику приложения. Модель должна быть

полностью независима от остальных частей продукта. Модельный слой ничего не должен знать об элементах дизайна, и каким образом он будет отображаться. Достигается результат, позволяющий менять представление данных, то как они отображаются, не трогая саму модель. Таким образом можно определить модель, как бизнес-логику приложения, которая обладает знаниями о себе самой и не знает о контроллерах и представлениях.

Для некоторых проектов модель — это просто слой данных (DAO, база данных, XML-файл), для других проектов — это менеджер базы данных, набор объектов или просто логика приложения.

В обязанности представления входит отображение данных полученных от модели. Однако, представление не может напрямую влиять на модель. Можно говорить, что представление обладает доступом «только на чтение» к данным. В представлении реализуется отображение данных, которые получаются от модели любым способом. Лишь в исключительных случаях представление может иметь код, который реализует некоторую бизнес-логику.

В свою очередь контроллер определяет, какие представление должно быть отображено в данный момент. Контроллер получает ввод пользователя, обрабатывает его и посылает обратно результат обработки, например, в виде представления. События представления могут повлиять только на контроллер. Контроллер может повлиять на модель и определить другое представление. Для одного контроллера возможно существование нескольких представлений.

В данном паттерне модель не зависит от представления или управляющей логики, что делает возможным проектирование модели как независимого компонента и, например, создавать несколько представлений для одной модели. Поэтому при использовании шаблона MVC можно быстро изменить способ представления данных, никак не затрагивая при этом логику работы приложения [10].

Сегодня MVC играет основополагающую роль в большинстве систем с пользовательским интерфейсом. Большинство фреймворков для веб-программирования, создания приложений для мобильных платформ содержат в своей основе содержат именно шаблон MVC.