

определения диагноза. Полносвязный и выходной слои отвечают за классификацию изображения и выдают конечный результат.

Разработка велась на языке программирования Python. В программе использовались библиотеки numpy, matplotlib, pandas и Tensorflow. Данные используемые для обучения сети были размечены и разделены на тестируемые и тренировочные.

Также для обучения и корректировки данной нейронной сети был использован алгоритм обратного распространения ошибки, для точной корректировки между всеми весами связей во всех слоях. Основная идея заключается в проверке состояния всех нейронов, вычислении ошибки для выходного слоя, проведении обратного распространения от конца к началу и итоговом обновлении весов [1].

В качестве функции активации используется ReLu так как при ее использовании не происходит затухание градиента, она не производит ресурсоемких операций, обучение сети будет происходить быстрее. Также при отрицательном выходе способна отсеять ненужные детали в признаках. Из минусов можно заметить довольно сильную зависимость от инициализации весов.

Одной из проблем возникших при разработке была стандартизация снимков к единому образцу при всех возможных методах проведения рентгеновского снимка головы и челюсти. Снимки могли быть проведены спереди сбоку под углом в зависимости от вида полученных травм так же они могли различаться по цветовой насыщенности и насыщенности цвета и резком снижении качества изображений.

При начальных тестированиях были достигнуты результаты верности постановки диагноза в 76,7% и 80,3%. Сверточные нейронные сети имеют хорошую устойчивость к явным искажениям изображений.

Заключение. Таким образом, использование сверточных нейронных сетей при решении задач прогнозирования диагноза с учетом входного, сверточного, подвыборочного, полносвязного и выходного слоев, является актуальным и эффективным для постановки диагноза.

1. Сверточная нейронная сеть. – Режим доступа: <https://habr.com/ru/post/348028/>. - Дата доступа: 15.03.2022.

2. Сверточная нейронная сеть. – Режим доступа: https://ru.wikipedia.org/wiki/Свёрточная_нейронная_сеть. – Дата доступа: 18.03.2022.

LAMBDA ARCHITECTURE ДЛЯ КОРПОРАТИВНОГО «ОЗЕРА ДАННЫХ»

Безворотных А.В.,

студентка 4 курса Сибирского федерального университета,

г. Красноярск, Российская Федерация

Научный руководитель – Кузьмич Р.И., канд. техн. наук, доцент

В современном производстве большую роль играют цифровые технологии, которые быстро внедряются во все отрасли производства.

Основной проблемой, возникающей при работе с данными, является обработка больших массивов данных в реальном времени, для решения которой в качестве инструмента можно использовать технологию «Озера данных» (data lake).

«Озеро данных» – интеллектуальное и технологическое связующее звено между источниками данных и потребителями информации, а также ядро обработки больших данных, которое позволяет преобразовывать разнородные необработанные источники данных в целевую информацию (рисунок 1).

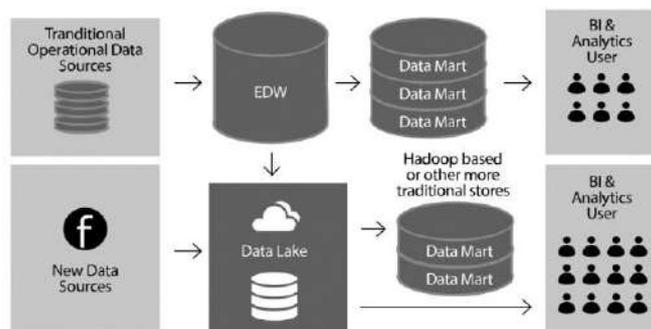


Рисунок 1 – Data Lake в качестве основного хранилища данных

В настоящее время использование технологии «Озера данных» на производстве характеризуется следующими преимуществами [1]:

- 1) максимальная гибкость технологии – ничего не нужно определять заранее;
- 2) более низкая стоимость – вы платите исключительно за занятые гигабайты;
- 3) подходит для создания большой корпоративной базы данных для масштабной аналитики.

Цель работы – исследовать архитектуру lambda для дальнейшего использования при проектировании корпоративного «Озера данных».

Материал и методы. Материалами исследования являются научные работы, связанные с технологией проектирования «Озера данных» на базе архитектуры lambda. В работе используются методы анализа и обобщения.

Результаты и их обсуждение. Архитектура lambda – представляет собой архитектуру конвейерного типа, предназначенную для снижения сложности конвейера аналитики в реальном времени с помощью ограничения инкрементных вычислений только небольшой частью этой архитектуры (рисунок 2).

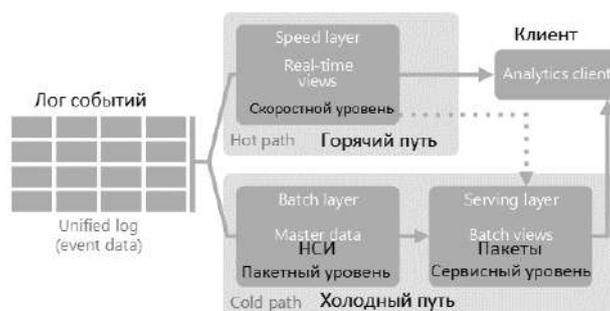


Рисунок 2 – Архитектура lambda

С помощью данного подхода обработки данных появляется возможность работы с классическим оперативным диапазоном (месяцы, часы) и с историческими данными, которые необходимо хранить в долгосрочной перспективе.

Существует два вида обработки: на пакетном уровне (холодный путь) и на уровне ускорения (критический путь) [2].

1. Критический путь, или уровень скорости, используется для быстрого приема чувствительных к задержке (или в реальном времени) данных, предназначенных для быстрого потребления клиентами аналитики.

2. Холодный путь, или пакетный слой, хранит все входящие данные для пакетной обработки, где обработка данных может занять от нескольких минут до нескольких часов.

В результате обработки критический и холодный пути объединяются в клиентском приложении аналитики. Для отображения результатов в режиме реального времени,

клиент будет получать результаты из критического пути. Другими словами, критический путь содержит данные за относительно небольшой промежуток времени, после которого результаты можно обновить более точными данными из холодного пути.

Необработанные данные, которые хранятся на пакетном уровне, являются неизменяемыми. Входящие данные всегда добавляются к имеющимся. Предыдущие данные никогда не перезаписываются. Любые изменения значения определенных данных хранятся в виде новой записи о событии с меткой времени. Это позволяет в любой момент времени выполнить повторное вычисление в журнале собранных данных. Возможность повторного вычисления пакетных представлений из исходных необработанных данных очень важна, так как это позволяет создавать представления по мере развития системы [3].

Заключение. Архитектура lambda для корпоративного «Озера данных» при правильном использовании позволяет справиться с обработкой и хранением увеличивающихся объемов данных в реальном времени и сократить объем инвестиций в классическое хранилище.

1. Иванов, П.Д. Технологии Big Data и их применение на современном промышленном предприятии [Электронный ресурс] / П.Д. Иванов, В.Ж. Вампилова // Инженерный журнал: Наука и инновации, 2014. Вып. 8. – Режим доступа: <http://engjournal.ru/catalog/it/asu/1228.html>

2. Гребнев, Е. Облачные сервисы: взгляд из России/ Е. Гребнев. – Издательство: Сnews 2011. – С. 282.

3. Клементьев, И. П., Введение в Облачные вычисления/ И.П. Клементьев, В.А Устинов. – Издательство: УГУ, 2009. – С. 223.

ГЕОГЕБРА – ПРИЛОЖЕНИЕ ДЛЯ ИЗУЧЕНИЯ МАТЕМАТИКИ

Гаджиева Ф.С., Молодечкин Н.А.,

магистранты 2 курса ВГУ имени П.М. Машерова, г. Витебск, Республика Беларусь

Научный руководитель – Ализарчик Л.Л., канд. пед. наук, доцент

В условиях информатизации образования педагоги в своей профессиональной деятельности совершенствуют методы преподавания математики, отвечая на вопросы: как модернизировать учебные занятия для повышения их эффективности; какое программное обеспечение лучше всего использовать в ходе урока для повышения мотивации современных учащихся и создания комфортных условий для творчества.

Электронные образовательные ресурсы для изучения математики должны быть удобными в использовании и обладать широким спектром функциональных возможностей. Одним из эффективных образовательных ресурсов является кроссплатформенное динамическое математическое приложение GeoGebra, которое может использоваться как средство визуализации информации, а также организации самостоятельной и творческой работы учащихся на занятиях по математике [1].

Цель работы – исследовать и апробировать функциональные и дидактические возможности приложения GeoGebra при изучении курса математики в учреждениях общего среднего образования [2, с. 9].

Материал и методы. В исследовании в качестве рабочего материала используется математическое приложение GeoGebra. Педагогический эксперимент проводился на базах ГУО «Гимназия №8 г. Витебска», ВГУ имени П.М. Машерова и Витебского областного института развития образования (ВОИРО).

Результаты и их обсуждение. В ходе исследования определены основные функциональные возможности приложения GeoGebra: создание сложных математических моделей и проведение компьютерных экспериментов.

С помощью данного программного обеспечения можно проводить построения графиков функций, заданных аналитически и параметрически, выполнять различного рода геометрические построения в двумерном и трехмерном пространствах, визуализировать этапы построения чертежа, вычислять производные и интегралы, факторизовать