

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

«ПЕНЗЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

Факультет
вычислительной техники

Кафедра
«Вычислительная техника»

Направление подготовки: 09.04.01 «Информатика и вычислительная техника»

Магистерская программа: «Прикладной искусственный интеллект»

МАГИСТЕРСКАЯ ДИССЕРТАЦИЯ

на тему

ПАРАЛЛЕЛЬНЫЕ И 3-Х МЕРНЫЕ
НЕЙРОННЫЕ СЕТИ НА ПРИМЕРЕ
РАСПОЗНОВАНИЯ РУКОПИСНЫХ ЦИФР

Студент

14.06.24 Мусонов Виталий Михайлович
(подпись, дата) (ФИО полностью)

Руководитель

14.06.24 Б.Д. Шашков
(подпись, дата) (фамилия, инициалы)

Нормоконтролёр

14.06.24 А.С. Бычков
(подпись, дата) (фамилия, инициалы)

Рецензент

14.06.24 А.Ю. Афонин
(подпись, дата) (фамилия, инициалы)

Работа допущена к защите (протокол зас. кафедры от 24.05.24 № 11)

Заведующий кафедрой

14.06.24 М.А. Митрохин
(подпись) (фамилия, инициалы)

Работа защищена с отметкой отлично (протокол заседания ГЭК от

18.06.2024 № 30)

Секретарь ГЭК

18.06.2024 Е.И. Гурин
(подпись) (фамилия, инициалы)

Пенза 2024

Федеральное государственное бюджетное образовательное учреждение
высшего образования
«ПЕНЗЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

Факультет
вычислительной техники

Кафедра
«Вычислительная техника»

«Утверждаю»
Заведующий кафедрой
М.А. Митрохин
« 13 » 09 2022г.

ЗАДАНИЕ
НА МАГИСТЕРСКУЮ ДИССЕРТАЦИЮ

1. Студент Мухомов Виталий Михайлович гр. 22ВВим2 факультета ВТ

2. Тема работы Параллельные и 3-х мерные нейронные сети на примере распознавания рукописных цифр

Тема утверждена приказом ПГУ № 765/о от 13 " 09 2022 г.

3. Руководитель магистерской диссертации к.т.н., профессор кафедры Шашков Б. Д.

4. Задание на работу (назначение разработки, исходные данные и т.п.)

Назначение разработки: исследование архитектур нейронных сетей для распознавания изображений, составление рекомендаций по использованию.

Создать и обучить нейронные сети с различными архитектурами. Собрать и проанализировать полученные результаты. Составить рекомендации по использованию.

5. Перечень подлежащих разработке вопросов

1) Обзор и анализ существующих нейронных сетей для распознавания изображений

2) Выбор подходов создания архитектуры для нейронных сетей для распознавания изображений

3) Разработка нейронных сетей для сравнения

4) Получение, сравнение, обработка, визуализация метрик, собранных во время обучения и тестирования нейронных сетей

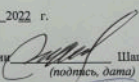
5) Подтверждение полученных данных

6) Составление рекомендаций по использованию

6. Календарный график выполнения работы

№ п/п	Наименование этапов работы	Объем работы	Срок выполнения	Подпись руководителя
1	Обзор и анализ существующих нейронных сетей для распознавания изображений	15%	16.09.2022-16.12.2022	
2	Выбор подходов создания архитектуры для нейронных сетей для распознавания изображений	15%	17.10.2022-18.02.2023	
3	Разработка нейронных сетей для сравнения	20%	19.02.2023-24.05.2023	
4	Получение, сравнение, обработка, визуализация метрик, собранных во время обучения и тестирования нейронных сетей	20%	25.05.2023-27.08.2023	
5	Подтверждение полученных данных	20%	28.08.2023-28.11.2023	
6	Составление рекомендаций по использованию	10%	29.11.2023-25.01.2024	

Дата выдачи задания " 13 " 09 2022 г.

Руководитель магистерской диссертации  Шашков Б. Д.
(подпись, дата)

Задание к исполнению принял студент  Мусонов В. М.
(подпись, дата)

Работу к защите допустить
Декаан факультета  Фионова Л. Р.
(подпись, дата)
19.06.2024

Реферат

Пояснительная записка содержит 61 лист, 15 таблицы, 22 рисунка.

НЕЙРОННЫЕ СЕТИ, ГЛУБОКОЕ ОБУЧЕНИЕ, ОБУЧЕНИЕ НЕЙРОННЫХ СЕТЕЙ, УСКОРИТЕЛИ НЕЙРОННОГО ОБУЧЕНИЯ, АРХИТЕКТУРЫ НЕЙРОННЫХ СЕТЕЙ, ИСКУССТВЕННЫЙ ИНТЕЛЛЕКТ

Объектом исследования являются архитектуры нейронных сетей.

Предметом исследования являются возможности оптимизации точности и скорости работы нейронной сети за счет модификации архитектуры.

Целью данной работы - исследование в направлении повышения производительности или точности нейронных сетей, а также составлении рекомендаций по использованию.

В результате проделанной работы был проведён анализ существующих архитектур нейронных сетей, выбор оптимальных архитектур, подходов и их компонент, созданы и обучены нейронные сети для проведения сравнения, собраны метрики для проведения сравнения, произведено сравнение, визуализация данных для сравнения, а также представлены рекомендации по применению обозреваемых подходов.

Научная новизна состоит в предложении использования однородного внутрисетевого параллелизма для оценки возможностей обучения нейронной сети. Структуризация применения архитектур нейронных сетей для распознавания изображений.

					ПГУ1.090401.40.001 ПЗ			
Изм	Имя	№ докум.	Подпись	Дата	Параллельные и 3-х мерные нейронные сети на примере распознавания рукописных цифр Пояснительная записка	Лит.	Лист	Листов
Разраб.	Мухомов В.М.						2	61
Провер.	Шашков Б.Д.					ФВТ, каф. ВТ гр. 22ВВИм2		
Н. Контр.	Бычков А.С.							
Утверд.	Митрохин М.А.							

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

«ПЕНЗЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

Факультет

вычислительной техники

Кафедра

«Вычислительная техника»

Направление подготовки: 09.04.01 «Информатика и вычислительная техника»

Магистерская программа: «Прикладной искусственный интеллект»

МАГИСТЕРСКАЯ ДИССЕРТАЦИЯ

на тему

ПАРАЛЛЕЛЬНЫЕ И 3-Х МЕРНЫЕ НЕЙ-
РОННЫЕ СЕТИ НА ПРИМЕРЕ РАСПОЗНО-
ВАНИЯ РУКОПИСНЫХ ЦИФР

Студент

Мусонов Виталий Михайлович

(подпись, дата)

(ФИО полностью)

Руководитель

Б.Д. Шашков

(подпись, дата)

(фамилия, инициалы)

Нормоконтролёр

А.С. Бычков

(подпись, дата)

(фамилия, инициалы)

Рецензент

(должность, место работы)

А.Ю. Афонин

(подпись, дата)

(фамилия, инициалы)

Работа допущена к защите (протокол зас. кафедры от _____ № 13)

Заведующий кафедрой

(подпись)

М.А. Митрохин

(фамилия, инициалы)

Работа защищена с отметкой _____ (протокол заседания ГЭК от

№ _____)

Секретарь ГЭК

(подпись)

Е.И. Гурин

(фамилия, инициалы)

Пенза 2024

Федеральное государственное бюджетное образовательное учреждение
высшего образования
«ПЕНЗЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

Факультет
вычислительной техники

Кафедра
«Вычислительная техника»

«Утверждаю»
Заведующий кафедрой
М.А. Митрохин
« 16 » 09 2022г.

**ЗАДАНИЕ
НА МАГИСТЕРСКУЮ ДИССЕРТАЦИЮ**

1. Студент Мусонов Виталий Михайлович гр. 22ВВИм2 факультета ВТ

2. Тема работы Параллельные и 3-х мерные нейронные сети на примере распознавания рукописных цифр

Тема утверждена приказом ПГУ № 765/о от " 16 " 09 2022 г.

3. Руководитель магистерской диссертации к.т.н., профессор кафедры Шашков Б. Д.

4. Задание на работу (назначение разработки, исходные данные и т.п.)

Назначение разработки: исследование архитектур нейронных сетей для распознавания изображений, составление рекомендаций по использованию.

Создать и обучить нейронные сети с различными архитектурами. Собрать и проанализировать полученные результаты. Составить рекомендации по использованию.

5. Перечень подлежащих разработке вопросов

1) Обзор и анализ существующих нейронных сетей для распознавания изображений

2) Выбор подходов создания архитектуры для нейронных сетей для распознавания изображений

3) Разработка нейронных сетей для сравнения

4) Получение, сравнение, обработка, визуализация метрик, собранных во время обучения и тестирования нейронных сетей

5) Подтверждение полученных данных

6) Составление рекомендаций по использованию

6. Календарный график выполнения работы

№ п/п	Наименование этапов работы	Объем работы	Срок выполнения	Подпись руководителя
1	Обзор и анализ существующих нейронных сетей для распознавания изображений	15%	16.09.2022-16.12.2022	
2	Выбор подходов создания архитектуры для нейронных сетей для распознавания изображений	15%	17.10.2022-18.02.2023	
3	Разработка нейронных сетей для сравнения	20%	19.02.2023-24.05.2023	
4	Получение, сравнение, обработка, визуализация метрик, собранных во время обучения и тестирования нейронных сетей	20%	25.05.2023-27.08.2023	
5	Подтверждение полученных данных	20%	28.08.2023-28.11.2023	
6	Составление рекомендаций по использованию	10%	29.11.2023-25.01.2024	

Дата выдачи задания " 16 " 09 _____ 2022 г.

Руководитель магистерской диссертации _____ Шашков Б. Д.
(подпись, дата)

Задание к исполнению принял студент _____ Мусонов В. М
(подпись, дата)

Работу к защите допустить
Декан факультета _____ Фионова Л. Р.
(подпись, дата)

Реферат

Пояснительная записка содержит 61 лист, 15 таблицы, 22 рисунка.

НЕЙРОННЫЕ СЕТИ, ГЛУБОКОЕ ОБУЧЕНИЕ, ОБУЧЕНИЕ НЕЙРОННЫХ СЕТЕЙ, УСКОРИТЕЛИ НЕЙРОННОГО ОБУЧЕНИЯ, АРХИТЕКТУРЫ НЕЙРОННЫХ СЕТЕЙ, ИСКУССТВЕННЫЙ ИНТЕЛЛЕКТ

Объектом исследования являются архитектуры нейронных сетей.

Предметом исследования являются возможности оптимизации точности и скорости работы нейронной сети за счет модификации архитектуры.

Целью данной работы - исследование в направлении повышения производительности или точности нейронных сетей, а также составлении рекомендаций по использованию.

В результате проделанной работы был проведён анализ существующих архитектур нейронных сетей, выбор оптимальных архитектур, подходов и их компонент, созданы и обучены нейронные сети для проведения сравнения, собраны метрики для проведения сравнения, произведено сравнение, визуализация данных для сравнения, а также представлены рекомендации по применению обзореваемых подходов.

Научная новизна состоит в предложении использования однородного внутрисетевого параллелизма для оценки возможностей обучения нейронной сети. Структуризация применения архитектур нейронных сетей для распознавания изображений.

					ПГУ1.090401.40.001 ПЗ			
Изм	Имя	№ докум.	Подпись	Дата				
Разраб.	Мусонов В.М				Параллельные и 3-х мерные нейронные сети на примере распознавания руко- писных цифр Пояснительная записка	Лит.	Лист	Листов
Провер.	Шашков Б.Д.							61
Н. Контр	Бычков А.С.					ФВТ, каф. ВТ		
Утверд.	Митрохин М.А.					гр. 22ВВИм2		

СОДЕРЖАНИЕ

ВВЕДЕНИЕ.....	3
1 Анализ предметной области.....	4
1.1 История развития архитектур нейронных сетей.....	4
1.2 Устройство нейронных сетей.....	9
1.3 Обучение нейронной сети.....	12
1.3.1 Обучение с учителем.....	13
1.3.2 Обучение без учителя.....	14
1.3.3 Обучение с подкреплением.....	14
2 Полносвязные сети.....	16
2.1 Разделение данных для полносвязных сетей.....	16
2.2 Проверка разделения данных для полносвязных сетей на более сложном наборе данных.....	23
2.3 Проверка разделения данных для полносвязных сетей. Вывод.....	27
3 Разделение данных в многомерных сетях.....	28
3.1 MNIST-digits.....	29
3.2 MNIST-fashion.....	30
3.3 Проверка разделения данных для сверточных сетей. Вывод.....	33
4 Параллелизм.....	35
4.1 Внесетевой параллелизм.....	35
4.1.1 Ансамблевое обучение.....	35
4.1.2 Слияние нейронных сетей.....	37
4.1.3 Выбор подхода.....	37
4.2 Внутрисетевой параллелизм.....	38
4.2.1 Неоднородный параллелизм.....	39
4.2.2 Однородный параллелизм.....	41
4.3 Варианты использования.....	50
ЗАКЛЮЧЕНИЕ.....	56
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ.....	60

ВВЕДЕНИЕ

С развитием систем обработки информации растет потребность в использовании нейронных сетей. Сети применяются во многих областях: программирование, дизайн, маркетинг, безопасность и других. Важными характеристиками нейронных сетей являются точность, скорость работы и время обучения.

Точность нейронной сети — первична. Точность — доля правильных ответов нейронной сети. Скорость работы нейронной сети важна в ряде задач. Например, оценка дорожной ситуации беспилотным транспортом. В представленном примере важна как скорости работы, так и точность. Это обосновывается, необходимостью быстро принимать решение, в соответствии с дорожной ситуацией, где цена промедления — человеческая жизнь. В остальных ситуациях нейронная сеть менее ограничена во времени обработки информации для предоставления ответа. Как пример — "рисование" изображения по запросу. Обработка пользовательских требований и само создание изображения ресурсоемкий процесс. В следствии, разработчик стремится к оптимизации потребляемых нейронной сетью ресурсов, увеличению итоговой точности сети. На указанные параметры влияет архитектура нейронной сети.

Цель работы заключается в исследовании в направлении повышения производительности или точности нейронных сетей, а также составлении рекомендаций по использованию.

1 Анализ предметной области

1.1 История развития архитектур нейронных сетей

Для задач машинного зрения изначально применялись полносвязные нейронные сети [1]. Изображение сворачивалось в одномерный список и подавалось на вход нейронной сети, где передавалось по слоям нейронной сети до выходного слоя. Нелинейность достигалась за счет использования в слоях сложных, нелинейных функций активации и их сочетанию.

Следующим шагом стало использование сверточных нейронных сетей. Задачей сверточной сети стало формирование новых признаков из изображения. Эти признаки получили название — карты признаков. Сверточные сети лучше подходят для изображений за счет особенностей в структуре. Особенности структуры в использовании ядер свертки. Присутствует зависимость между количеством ядер и количеством выходных признаков. Эти параметры указываются программистом. В цветных изображениях применяются каналы, представляющие из себя матрицу, где каждый элемент — значение цвета пикселя. Для каждого канала используется один цвет. Работа с каналами также структурно предусмотрена в сверточных сетях. При применении операции свертки происходит извлечение карты признаков. Карта признаков — массив матриц, где каждый канал отвечает за извлеченный признак. В основе свертки лежит ядро свертки — матрица с заданным программистом размером. Ядро свертки накладывается на левый верхний угол изображения и производится покомпонентное умножение значений фильтра и значений матрицы изображения под фильтром. После этого действия фильтр сдвигается на шаг, указанный программистом. По достижении правой границы изображения происходит возвращение к левой границе и ядро свертки сдвигается в направлении нижней границы изображения на значение, указанное программистом. При необходимости возможно исключить обязательность сдвига к левой части изображения. Эти действия повторяются до достижения ядром свертки нижнего угла изображения справа.

Далее сети стали усложняться и для их представления можно использовать структуру графа. Где элемент графа — слой нейронной сети. За счет введения графовой структуры также возникает смысл оптимизации скорости работы нейронной сети за счет использования нескольких вычислителей. Так как, теперь, стало возможно использовать параллельные вычисления. Так же за счет использования этой структуры появляются потоки специфично обработанных данных, так как в каждой ветви используются иные наборы слоев. Специфичность обработки заключается в использовании сочетаний настроенных слоев разными способами и тонкой настройки каждой ветви во время разработки. За счет усложнения обработки информации увеличилась точность работы итоговой сети. Так как происходят модификации большим количеством сложных обработок без взаимного влияния.

Важным этапом стал ранний выход из нейронной сети. За счет использования графовой структуры стало возможно устанавливать больше одной выходной точки из нейронной сети. Это решение позволило сократить время обработки, за счет неполного прохождения информации по сети. Этот подход возможно применять на задачах классификации по принципу от большего к меньшему, где большие классы («крупные» детали на изображениях) могут классифицироваться без полного прохода сети. Что ведет к большей скорости работы классификатора.

В аппаратной части тоже произошли изменения. Обучать и эксплуатировать нейронные сети стали с использованием графических процессоров и специальных нейронных ускорителей. Раньше обучали и запускали нейронные сети только на центральном процессоре компьютера. Компании используют нейроморфные процессоры как сопроцессоры в производимых устройствах (например Huawei). Разработчики нейроморфных процессоров выделяют преимущества использования этих процессоров: сокращение времени обучения нейронной сети, уменьшение физических размеров устройств и меньшее энергопотребление.

В начале, для обучения нейронных сетей, применялись центральные процессоры компьютеров. Нейронные сети становились больше. При увеличении количества нейронов в одном слое стало труднее разделять подсчет при использовании ядер процессора, так как количество нейронов в одном слое стало больше, чем доступных ядер процессора. Это стало причиной поиска новой аппаратной базы для создания нейронных сетей за меньшее время.

В 2004 начали использовать мощности графического процессора [2]. Графические карты быстро стали заменой центральным процессорам, за счет большего количества вычислительных ядер. Также при увеличении количества ядер упростилась параллелизация нейронных сетей. Вследствие этого время обучения уменьшилось, а скорость работы нейронных сетей увеличилась. В 2006 году компания Nvidia представила программно-аппаратный комплекс разработки — CUDA. Это позволило упростить написание кода, повысило эффективность работы, так как для этой технологии установлены одноименные ядра, с большей скоростью работы, ориентированные на параллельные вычисления. В 2016 году аналогичную технологию представила компания AMD — ROCm. ROCm и CUDA отличаются. Исходный код ROCm доступен для изменения сообществом программистов, но эта технология менее производительна. Меньшая производительность обусловлена технологическим отставанием компании AMD.

В 2015 году компанией Google представлены тензорные процессоры. Тензорные ядра работают быстрее чем CUDA-ядра за счёт меньшего количества циклов для операций с матрицами [3]. Также, при необходимости возможно перейти к режиму пониженной точности и обучать вдвое большую модель за счет использования 16 бит вместо 32 для кодирования чисел с плавающей точкой. Важно при этом учитывать возможное изменение точности работы сети.

Для узких задач реализовали специальные процессоры. Например, процессоры машинного зрения, предназначенные для аппаратного ускорения алгоритмов машинного зрения.

Также разработали программные комплексы с поддержкой аппаратных технологий. Эти комплексы упрощают подготовку данных, обучение и запуск нейронных сетей. Популярные программные комплексы — TensorFlow, PyTorch, Keras, Caffe [4]. Каждый из представленных комплексов отличается. Так Caffe поддерживает только работу с языком C++.

Каждая система обладает как достоинствами, так и недостатками [5]. Сравнивая TensorFlow и PyTorch выявляются различия:

- TensorFlow больше распространена и как следствие возможно найти большее количество ресурсов,
- PyTorch применяет динамическое определение графа,
- PyTorch позволяет проводить отладку отладчиком языка Python в то время, как TensorFlow требует дополнительного инструмента отладки,
- TensorFlow предоставляет отдельный, мощный инструмент визуализации — TensorBoard,
- TensorFlow может разворачиваться на большом количестве устройств различными способами.

В работе рассматривается задача распознавания образов и поэтому рассматриваются сети связанные с решением этой задачи. Для исследований и модификаций выбирается сеть LeNet-5. Для исследования архитектурных особенностей, связанных с параллелизмом выбрана AlexNet.

LeNet-5 выбрана по причине простой, относительно других нейронных сетей, структуры. LeNet-5 сверточная сеть, слой-классификатор представлен в виде полносвязной многослойной нейронной сети [6]. Оригинальная архитектура LeNet-5 представлена на рисунке 1.

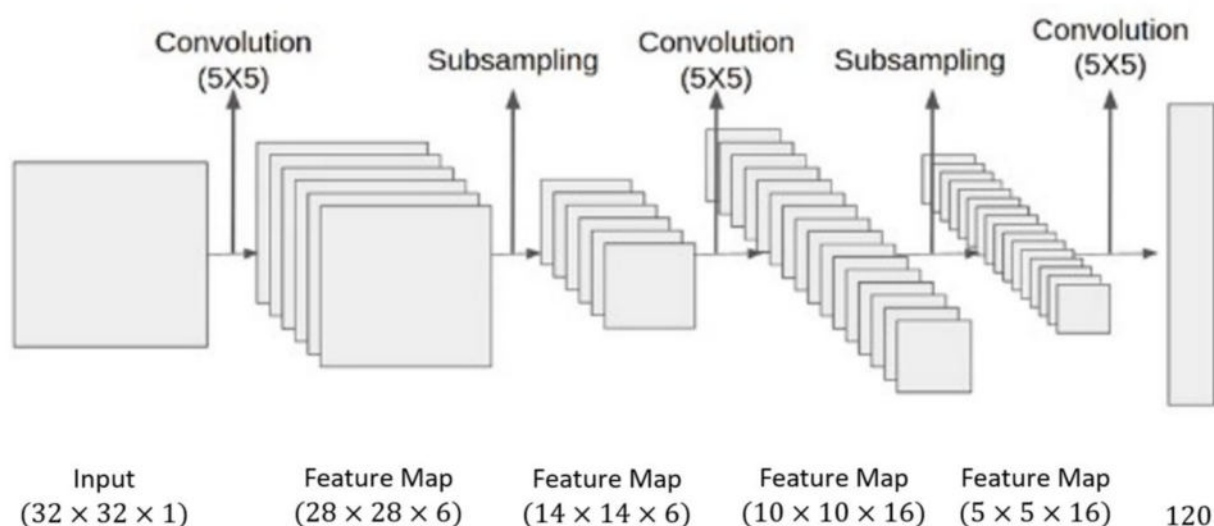


Рисунок 1 – Архитектура LeNet-5

AlexNet более сложная и точная чем LeNet-5 [7]. Извлечение признаков реализуется с использованием сверточных слоев. Также использует полносвязный слой-классификатор. Также есть слои для предотвращения проблемы переобучения. Использует функцию активации ReLU в слоях. Для экспериментов с изменением данных, но без изменения архитектуры предполагается использование AlexNet. Так как возможно использовать веса уже обученной сети. Архитектура AlexNet представлена на рисунке 2.

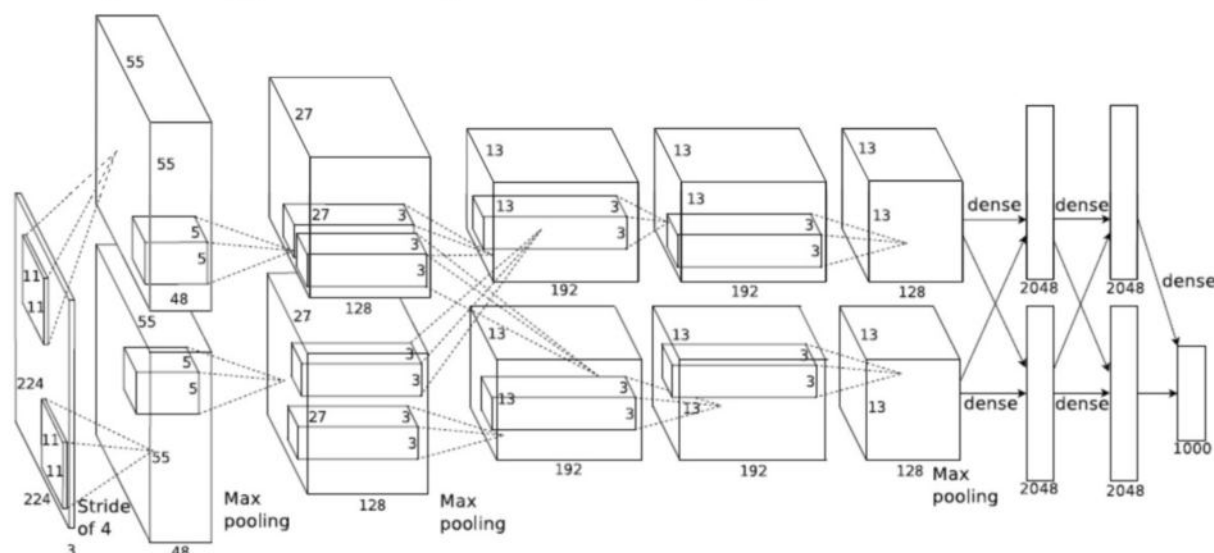


Рисунок 2 – Архитектура AlexNet

При создании AlexNet у авторов не было доступа к вычислителям, с достаточной мощностью. Поэтому авторы сети разделили сеть на две подсети.

Обе подсети обучались одновременно на двух графических ускорителях. При этом обучение сети заняло шесть дней.

1.2 Устройство нейронных сетей

Искусственный нейрон – базовый элемент искусственной нейронной сети. Искусственный нейрон состоит из:

- входы в нейрон,
- вектора весов в количестве, равном количеству входов,
- функции суммы весов,
- функции активации,
- параметров функции активации.

На каждый вход нейрона подается значение. Каждое значение имеет свой индекс. При помощи этого индекса извлекается значение веса и умножается на входящее значение. После этого происходит все умноженные значения суммируются и подаются на вход функцию активации. Функция активации, используя собственные параметры и данные, переданные ей, возвращает итоговый результат. Схематичное изображение искусственного нейрона представлено на рисунке 3.

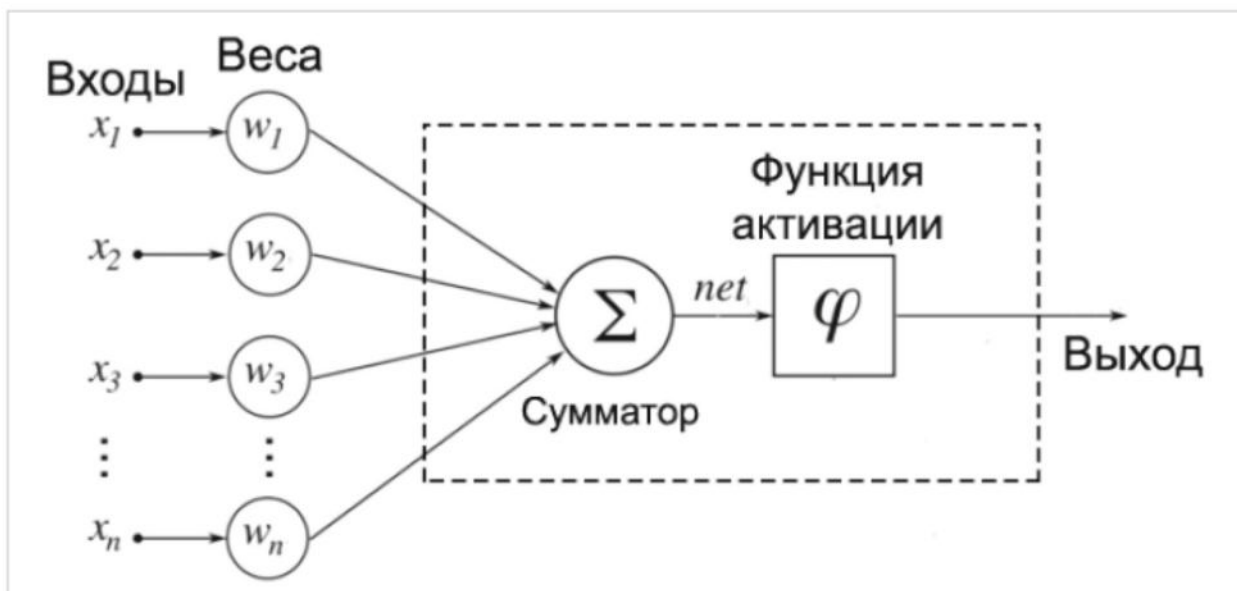


Рисунок 3 - Схематичное изображение искусственного нейрона

Из искусственных нейронов созданы полносвязные нейронные сети. Полносвязные нейронные сети состоят из слоев. В каждом слое размещается

некоторое количество нейронов. Количество нейронов определяется разрабатчиком сети. Обычно, нейроны внутри одного слоя не имеют связи друг с другом, но возможны реализации, в которых они будут соединены таким образом. Слои имеют названия: входной, скрытый, выходной. На входной слой подаются данные для обработки. На скрытых слоях обработка происходит. Выходной слой применяется для вывода результата работы нейронной сети. Скрытые слои располагаются между входным и выходным слоями. Если количество скрытых слоев более одного нейросеть считается глубокой, иначе – не глубокой. Для полносвязных нейронных сетей существует множество различных архитектур. Часть из них представлена на рисунке 4.

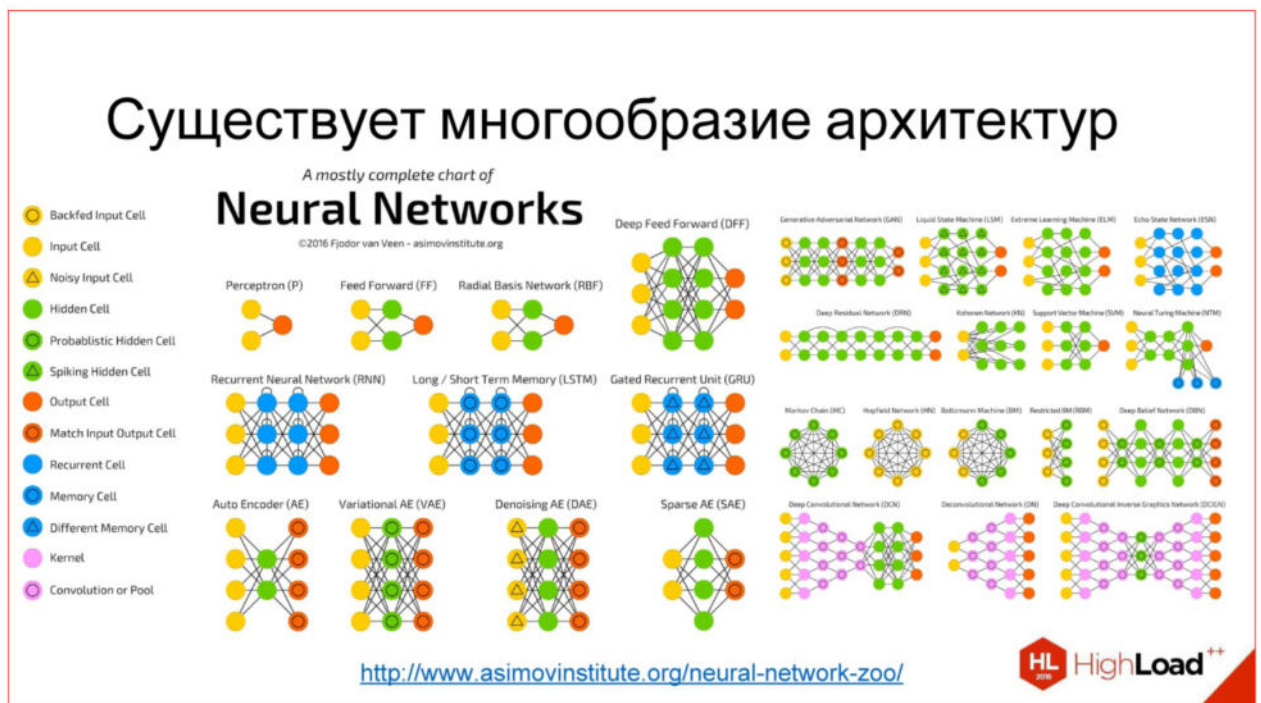


Рисунок 4 - Архитектуры полносвязных нейронных сетей

Каждая архитектура нейронной сети решает различные задачи. Например, сверточные нейронные сети лучше производят обработку изображений. Выбор сети зависит исключительно от решаемой задачи. При этом возможно комбинирование различных слоев для получения более точного результата.

Персептрон является одним из первых архитектур нейронной сети. В общем виде, предложенном Розенблатт, он представляет систему из трех типов элементов: сенсор, ассоциативный элемент и реагирующий элемент. Сен-

сены могут находиться в двух состояниях: покой, возбуждение. При взаимодействии с сенсором он возбуждается или остается в состоянии покоя. Сенсоры соединяются с ассоциативными элементами. Связи могут обладать весами, но веса могут иметь только конкретные значения: -1, 0, 1. К одному ассоциативному элементу могут быть подключено несколько сенсорных элементов. Ассоциативный элемент переходит в возбужденное состояние, если совокупное значение сигналов преодолевает порог срабатывания элемента. Возбужденное состояние ассоциативного элемента равно 1, иначе 0. После ассоциативного элемента следует реагирующий элемент. К нему также возможно подключение нескольких ассоциативных элементов. В случае превышения порога, установленного для реагирующего элемента, им выдается значение 1 иначе -1. Каждый элемент может существовать в более чем одном экземпляре. Схема персептрона представлена на рисунке 5.

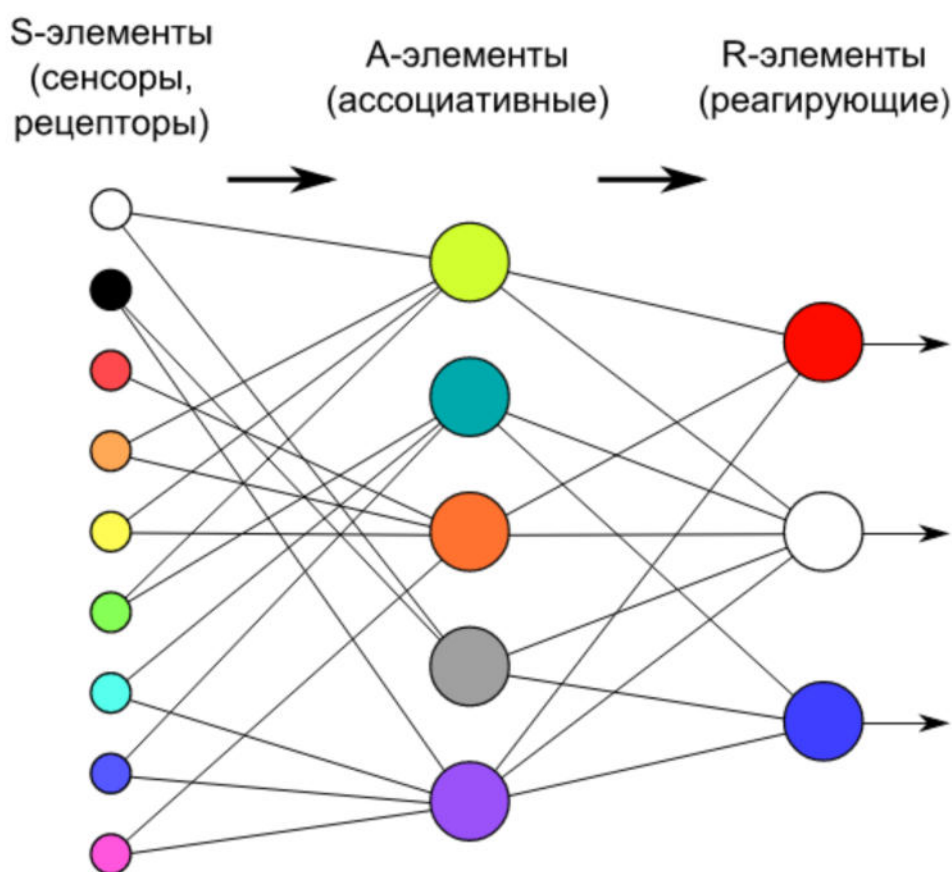


Рисунок 5 - Схема персептрона

Персептроны классифицированы как:

- однослойный персептрон,
- многослойный персептрон.

Однослойный персептрон – персептрон, каждый сенсор которого однозначно соответствует одному ассоциативному элементу, вес связи между ними всегда равен 1, порог любого ассоциативного элемента – 1.

Под многослойным персептроном подразумевается два разных вида:

- многослойный персептрон Румельхарта,
- многослойный персептрон Розенблатта.

По Румельзарту многослойный персептрон является частным случаем однослойного персептрона Розенблатта с особенностями:

- связи между сенсорами и ассоциативными элементами могут иметь произвольные веса и обучаться наравне с связями между ассоциативными элементами и реагирующими элементами,
- обучение производится согласно алгоритму обратного распределения ошибки.

По Розенблатту многослойный персептрон – это персептрон содержащий более одного слоя ассоциативных элементов.

1.3 Обучение нейронной сети

Процесс обучения нейрона заключается в нахождении правильного веса каждого нейрона. Возможна ситуация, при которой, невозможно найти веса, при котором нейронная сеть будет решать задачу с точностью в 100%. В таком случае возникает ошибка нейронной сети.

Существуют варианты обучения нейронной сети:

- обучение с учителем,
- обучение без учителя,
- обучение с подкреплением.

Первым этапом любого обучения является генерация начальных весов у нейронов. Для генерации начальных весов отвечает генератор случайных значений.

1.3.1 Обучение с учителем

Обучение с учителем требует наличие размеченного набора данных для тренировки модели нейронной сети. В зависимости от типа решаемой задачи разметка отличается. Например, для задачи классификации, необходимо иметь свойства объекта классификации и метку класса. Начальным этапом обучения с учителем является подготовка данных. Необходимо найти и подготовить набор входных данных для нейронной сети и необходимо иметь правильный ответ, соответствующий этим данным. Также необходимо разбить полученные данные на несколько категорий: тренировочные, тестовые, проверочные. Тренировочные данные используются для обучения нейронной сети. Тестовые данные используются для проверки нейронной сети во время обучения, проверочные – для проверки сети после процесса обучения. Обучение происходит поэтапно и итерационно. Обычно имеется два цикла: цикл эпох и цикл итераций. Итерации происходят в рамках эпохи. Во время эпохи, внутри итераций, через нейронную сеть проходит весь тренировочный набор данных. В рамках одной эпохи через нейронную сеть проходит часть данных – батч. Батч – фрагмент набора данных.

Во время итерации, данные подаются на вход нейронной сети. После ответа нейронной сети, он сравнивается с правильным. Во время сравнения ответов высчитывается значение ошибки прогноза сети. В случае отсутствия ошибки веса не изменяются и происходит переход к следующей итерации. В ином случае применяется алгоритм обратного распространения ошибки. В рамках этого алгоритма происходит оценка влияния нейронов на возникновение этой ошибки и корректировка веса. При оценке влияния используется функция потерь. Задачей оптимизации нейронной сети является минимизация функции потерь. Выбор этой функции зависит от конкретно решаемой задачи.

После процесса обучения происходит процесс, называемый валидацией. Во время этого процесса происходит финальная проверка результата процесса обучения. Важно, чтобы данные в проверочной выборке не пересекались в тестовых, тренировочных выборках. Так как, сеть уже сталкивалась с ними.

1.3.2 Обучение без учителя

Не всегда возможно найти и разметить данные. В таких случаях применяется обучение без учителя. Задача нейронной сети в таком подходе «самостоятельно» найти полезные признаки и произвести их анализ. Возможно частичное привлечение учителя. В таком случае часть данных будет размечена, а часть нет. Подход частичного привлечения учителя очень полезен, когда трудно извлечь важные признаки или разметить все объекты.

Один из методов обучения в таком случае – генеративно-состязательное. В этом случае существует две нейронных сети – генератор и дискриминатор. Генератор пытается создать данные, схожие с обучающей выборкой, дискриминатор оценивает данные и определяет являются ли сгенерированные данные поддельными. Таким образом данная связка постоянно совершенствуется. Дискриминатор точнее находит подделки, а генератор лучше их создает. Этот подход имеет большую популярность в нейронных сетях для генерации изображений.

1.3.3 Обучение с подкреплением

В данном виде обучения нейронная сеть управляет некоторым агентом в конкретной среде. Часто это применяется в видеоиграх. Агент имеет ограниченный набор возможных действий. Действия агента имеют некоторый результат. Действие или набор действий имеет результат, выраженный в виде оценки. Оценка может быть положительной или отрицательной. Задача обучения в данном случае – получение нейронной сетью максимально возможного количества положительных оценок. Процесс обучения с подкреплением также является итеративным процессом. В каждой итерации нейронная сеть управляет агентом в его окружении.

В качестве окружения может быть реальный или проявления виртуального. Сначала сеть действует случайным образом, но по мере обучения начинает выполнять больше поступков, которые приносят положительные оценки и совершает меньше отрицательных действий.

2 Полносвязные сети

Полносвязные нейронные сети — класс нейронных сетей, в котором каждый нейрон текущего слоя соединяется с каждым нейроном следующего слоя. Регулировка соединения строится за счет изменения веса соединения. Полносвязные нейронные сети подходят для классификации, регрессии, аппроксимации функций. Несмотря на возможные варианты использования представленный класс нейронных сетей обладает рядом проблем, к которым относятся [8]:

- вычислительная сложность, по причине большего количества параметров сети,
- переобучение, которое приводит к сходимости на этапе подготовки сети выше, чем во время проверки сети,
- меньшая точность в сравнении со сверточными нейронными сетями в работе с изображениями,
- чувствительность к входным данным, которая приводит к малой устойчивости к шумам и искажениям данных,
- ограниченность в работе с последовательными данными, такие как временные ряды или текст, так как полносвязные сети могут не учитывать зависимости между элементами последовательности.

Для задачи классификации изображений возникает дополнительная проблема — чувствительность к перемещению объектов на изображении. В ситуации, когда объекты на изображении находятся неизменно в одной и той же позиции на каждом изображении этой проблемой возможно пренебречь. В ситуации, в которой нельзя зафиксировать объект в одном и том же месте на изображении — точность будет снижаться.

2.1 Разделение данных для полносвязных сетей

За счет уменьшения изображения возможно использовать меньшее количество параметров. Также при уменьшении параметров сети ожидается снижение точности. При этом, чем меньше параметров обрабатывает нейронная сеть — тем выше ожидается скорость ответа нейронной сети, так как вычис-

лителю требуется меньше времени на обработку информации. В текущем разделе экспериментально проверяются эти тезисы. Для проверки используется набор данных MNIST-digits. В этом наборе содержатся прошедшие классификацию изображения рукописных цифр. Каждое изображение помечено принадлежностью к одному из десяти классов из набора данных.

Для решения задачи создана полносвязная нейронная сеть. В первую очередь подготавливается нейронная сеть, с которой производится сравнение остальных. Для эксперимента создана нейронная сеть с двумя скрытыми слоями. Первый слой выполняет получение входной информации. Последний слой выполняет вывод результата на основании работы скрытых слоев. Архитектура созданной сети демонстрируется на рисунке 6.

```
Net(  
  (fc1): Linear(in_features=49, out_features=98, bias=True)  
  (fc2): Linear(in_features=98, out_features=196, bias=True)  
  (fc3): Linear(in_features=196, out_features=98, bias=True)  
  (fc4): Linear(in_features=98, out_features=10, bias=True)  
)
```

Рисунок 6 – Архитектура полносвязной сети

Размеры каждого слоя зависят от размеров изображения для обработки. Первый слой данной сети — площадь изображения в пикселях. В первом скрытом слое в два раза больше нейронов чем в входном слое. Во втором скрытом слое нейронов в два раза больше, чем во втором слое. В последнем слое всего десять нейронов. Каждый нейрон в последнем слое означает класс изображения — цифра от 0 до 9.

Функцией активации выбирается ReLU. Причины выбора, следующие:

- устраняет проблему исчезающего градиента,
- требует меньше ресурсов для вычислений,
- при отрицательных значениях аргумента, значение функции — 0.

Проведен ряд экспериментов и получен список значений точности сетей на каждом из экспериментов. Произведено вычисление среднего арифмети-

ческого для нивелирования влияния генератора случайных значений на результат. Генератор случайных значений используется на начальном этапе обучения нейронной сети для первоначальной установки весов в связи между искусственными нейронами. Получен средний показатель точности работы сети — 0.97.

В следующем эксперименте уменьшается размер исходного изображения. Для следующего этапа эксперимента берется верхняя половина изображения. Дальнейшие этапы происходят аналогично подготовке исходной сети, где используются исходное изображение. Так как размер изображения становится меньше — сеть уменьшается. Изменение уменьшения вытекает из описания структуры сети. Получение и обработка результата точности сети построено аналогично обработке результатов работы исходной сети. Средняя, на ряде экспериментов точность, составила 0.96. Аналогичный эксперимент произведен с использованием только правой вертикальной частью изображения. В этом случае средняя точность составила 0.97.

В ситуации, где количество различий между классами на изображении велико — точность сети выше, чем в ином случае. В следующем эксперименте было сокращены высота и ширина изображения вдвое в сравнении с оригинальным изображением. Среднее значение точности, полученное в этом эксперименте, снизилось до 0.96. Пример обрезанных изображений с ответами сетей представлены на рисунке 7.

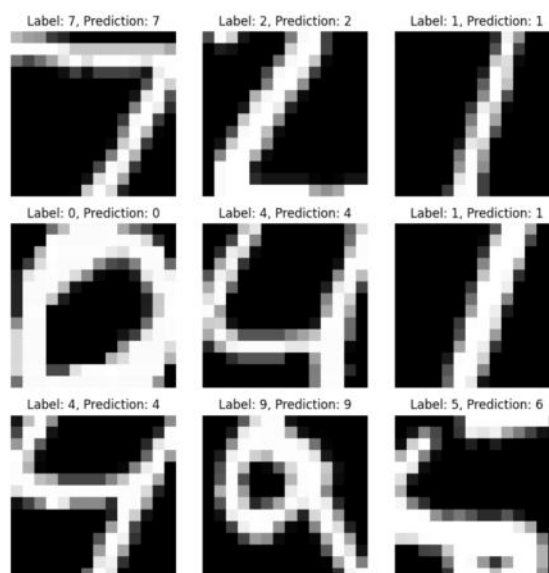


Рисунок 7 – Тестовые изображения с ответами нейронной сети

В последнем эксперименте изображение было уменьшено до размеров семь на семь пикселей. Ширина и высота изображения для текущего эксперимента составляют половину соответствующих параметров прошлого эксперимента. После проведения подсчета получено значение точности сети, обученной для этого эксперимента — 0.79.

Помимо точности, также собирались характеристики: время ответа нейронной сети, и время ее обучения. Значения получены и обработаны аналогично получению и обработки параметра точности нейронной сети и оформлены в рамках сводной таблицы. Сводная таблица представлена как таблица 1.

Таблица 1 – Результаты экспериментов манипуляции данных

Сеть	Точность	Время обучения (с)	Время от-вета (мс) ²	Высота изобра-жения (пиксел)	Ширина изображения (пиксел)
С полными данными	0,97	77,45	0,59	28	28
Обрезанная по горизонтали	0,96	76,34	0,58	14	28

Продолжение таблицы 1

Обрезанная по вертикали	0,97	76,28	0,58	28	14
Обрезанная по вертикали и горизонтали	0,96	75,50	0,58	14	14
Сильная обрезка	0,79	77,56	0,59	7	7

После создания сводной таблицы произведено перекрестное сравнение полученных в ходе эксперимента сетей по выделенным характеристикам. Перекрестное сравнение опирается на данные, представленные в сводной таблице. Для каждого перекрестного сравнения составлены таблицы. Сравнения представлены на таблицах 2-5.

Таблица 2 – Сравнение точности при модификации данных в полносвязных сетях

Сеть	С полными данными	Обрезанная по горизонтали	Обрезанная по вертикали и горизонтали	Обрезанная по вертикали и горизонтали	Сильная обрезка
С полными данными	###	0,00	0,00	0,01	0,19
Обрезанная по горизонтали	-0,01	###	-0,01	0,00	0,18

Продолжение таблицы 2

Обрезанная по вертикале и горизонтали	0,00	0,01	###	0,01	0,19
Обрезанная по вертикале и горизонтали	-0,01	0,00	-0,01	###	0,17
Сильная обрезка	-0,19	0,18	0,19	0,17	###

В сравнении точности обозреваемых нейронных сетей наблюдается зависимость между точностью нейронной сети и полнотой входного изображения. В используемом для обучения сетей наборе, изображения расположены по центру. Следовательно, при использовании частей изображения, содержащих его центр, нейронная сеть находит признаки достаточные для обобщения. Последняя сеть демонстрирует наименьшую точность из представленных. Обоснование заключается в меньшем доступном количестве признаков для оценки.

Таблица 3 – Сравнение скорости ответа при модификации данных в полносвязных сетях

Сеть	С полными данными	Обрезанная по горизонтали	Обрезанная по вертикали и горизонтали	Обрезанная по вертикали и горизонтали	Сильная обрезка
С полными данными	###	0,01	0,01	0,01	0,00
Обрезанная по горизонтали	-0,01	###	0,00	0,00	0,00

Продолжение таблицы 3

Обрезанная по вертикали и горизонтали	-0,01	0,00	###		-0,01
Обрезанная по вертикали и горизонтали	-0,01	0,00	0,00	###	-0,01
Сильная обрезка	0,00	0,00	0,01		###

При использовании набора данных MNIST-digits наблюдается малое отличие в скорости реакции нейронной сети.

Таблица 4 – Сравнение скорости обучения при модификации данных в полносвязных сетях

Сеть	С полными данными	Обрезанная по горизонтали	Обрезанная по вертикали и горизонтали	Обрезанная по вертикали и горизонтали	Сильная обрезка
С полными данными	###	1,11	1,17	1,94	2,35
Обрезанная по горизонтали	-1,11	###	0,06	0,83	1,24
Обрезанная по вертикали и горизонтали	-1,17	-0,06	###	0,83	1,24

Продолжение таблицы 4

Обрезанная по вертикали и горизонтали	-1,94	-0,83	-0,78	###	0,40
Сильная обрезка	-2,35	-1,24	-1,18	-0,40	###

При использовании меньших изображений время обучения сокращается.

2.2 Проверка разделения данных для полносвязных сетей на более сложном наборе данных

В ходе предыдущих экспериментов время ответа нейронной сети имело слишком малое изменение. Решено произвести повторный эксперимент. Отличие эксперимента — использование усложненного набора данных. Выбран набор данных MNIST-fashion. Этот набор предоставляется для обучения нейронных сетей задаче классификации и состоит из изображений одежды.

Произведено обучение сетей и обработка результатов аналогично прошлым экспериментам. Результаты усреднены и помещены в свободную таблицу 5.

Таблица 5 – Сводная таблица сравнения разделения данных для MNIST-fashion

Сеть	Точность	Время обучения (с)	Время ответа (мс) ²	Высота изображения	Ширина изображения (пиксел)
С полными данными	0,87	79,04	0,61	28	28

Продолжение таблицы 5

Обрезанная по горизонтали	0,81	78,74	0,61	14	28
Обрезанная по вертикали и горизонтали	0,83	79,81	0,66	28	14
Обрезанная по вертикали и горизонтали	0,73	83,00	0,64	14	14
Сильная обрезка	0,53	83,00	0,64	7	7

После создания свободной таблицы автор произвел перекрестное сравнение по характеристиками представленным в сводной таблице. Результаты перекрестного сравнения представлены в таблицах 6-8.

Таблица 6 – Сравнение точности работы между полученными сетями

Сеть	С полными данными	Обрезанная по горизонтали	Обрезанная по вертикали и горизонтали	Обрезанная по вертикали и горизонтали	Сильная обрезка
С полными данными	###	0,04	0,04	0,14	0,34
Обрезанная по горизонтали	-0,06	###	-0,02	0,08	0,28
Обрезанная по вертикали	-0,04	0,02	###	0,10	0,30

Продолжение таблицы 6

Обрезанная по вертикали и горизонтали	-0,14	-0,08	-0,10	###	0,19
Сильная обрезка	-0,34	-0,28	-0,30	-0,19	###

Сравнение точности с использованием изображений из набора MNIST-fashion подтверждает результаты сравнений с использованием набора MNIST-digits. Сравнение показывает постепенное уменьшение точности при количественном уменьшении признаков. Количество возможных признаков уменьшается из-за уменьшения размера изображения. Также в MNIST-fashion применяется центрирование изображения. Из-за большей сложности получения признаков на этом наборе, увеличивается риск недополучения признаков. Это демонстрируется при проведении сравнения между сетями, обрабатывающие вертикальные и горизонтальные половины изображения. Следовательно, в текущем наборе данных возможно извлечь больше признаков двигаясь вдоль высоты изображения.

Таблица 7 – Сравнение скорости ответа между полученными сетями

Сеть	С полными данными	Обрезанная по горизонтали	Обрезанная по вертикали и горизонтали	Обрезанная по вертикали и горизонтали	Сильная обрезка
С полными данными	###	0,00	-0,05	-0,02	-0,02
Обрезанная по горизонтали	0,00	###	-0,05	-0,03	-0,03

Продолжение таблицы 7

Обрезанная по вертикали и горизонтали	0,05	0,05	###		0,03
Обрезанная по вертикали и горизонтали	0,02	0,03	-0,03	###	0,00
Сильная обрезка	0,02	0,03	-0,03	0,00	###

В представленных экспериментах наблюдаются изменения скорости ответа нейронных сетей. Данное поведение планируется изучить в следующих работах.

Таблица 8 – Сравнение скорости обучения между полученными сетями

Сеть	С полными данными	Обрезанная по горизонтали	Обрезанная по вертикали и горизонтали	Обрезанная по вертикали и горизонтали	Сильная обрезка
С полными данными	###	0,30	-0,77	-3,96	-3,96
Обрезанная по горизонтали	-0,30	###	-1,08	-4,26	-4,26
Обрезанная по вертикали и горизонтали	0,77	1,08	###	-4,26	-4,26

Продолжение таблицы 8

Обрезанная по вертикали и горизонтали	3,96	4,26	3,18	###	0,00
Сильная обреза	3,96	4,26	3,18	0,00	###

Уменьшении размера входящего изображения выявлено уменьшение времени, необходимого на обучение сети.

2.3 Проверка разделения данных для полносвязных сетей.

Вывод

В исследуемом подходе наблюдается: понижение точности, сложное изменение скорости ответа и уменьшении времени обучения. Метод можно применить для сокращения обучаемых параметров нейронной сети. Но, при соблюдении условия нахождения достаточного количества признаков на фрагменте используемого изображения. Следовательно соблюдение условия зависит от решаемой задачи, используемого набора данных. При использовании полносвязных нейронных сетей этот подход не рекомендуется.

3 Разделение данных в многомерных сетях

В задачах распознавания образов и классификации изображений применяются сверточные нейронные сети. В сравнении с полносвязными нейронными сетями сверточные сети точнее. Обученные сверточные слои, реагируют на признаки, закрепленные во время обучения. Расположение признаков на изображении не важно.

Этот раздел повторяет эксперименты, описанные в предыдущем разделе. Отличия заключаются в слоях нейронной сети. Выбрана нейронная сеть LeNet-5. Эту сеть реализовать проще других сетей, рассматриваемых в текущей работе.

Выбранная сеть реализована. Архитектура сети представлена на рисунке 8.

```
class ConvNet(nn.Module):
    def __init__(self):
        super(ConvNet, self).__init__()
        self.layer1 = nn.Sequential(
            nn.Conv2d(1, 32, kernel_size=2, stride=1, padding=2),
            nn.ReLU(),
            nn.MaxPool2d(kernel_size=2, stride=2)
        )
        self.layer2 = nn.Sequential(
            nn.Conv2d(32, 64, kernel_size=5, stride=1, padding=2),
            nn.ReLU(),
            nn.MaxPool2d(kernel_size=2, stride=2)
        )

        self.drop_out = nn.Dropout()
        self.fc1 = nn.Linear(64 * 7 * 7, 1000)
        self.fc2 = nn.Linear(1000, 10)
        pass

    def forward(self, x):
        out = self.layer1(x)
        out = self.layer2(out)
        out = out.reshape(out.size(0), -1)
        out = self.drop_out(out)
        out = self.fc1(out)
        out = self.fc2(out)
        return out
```

Рисунок 8 – Архитектура сверточной сети для разделения данных

Архитектура LeNet-5 представляет из себя три модуля. Первый и второй модуль занимаются отбором признаков. В составе первого и второго модуля

применяются сверточные слои и слой MaxPooling. MaxPooling используется для уменьшения количества информации [9]. Принцип работы данного слоя схож с принципом работы сверточного слоя. Вместо покомпонентного умножения в MaxPooling происходит выборка максимального значения из матриц. Первый и второй модули отличаются параметрами сверточных слоев. Первый модуль извлекает 32 признака с размером ядра в 2 пикселя. Второй модуль извлекает 64 признака с размером ядра в 5 пикселей. Таким образом, отличие заключается в размере извлекаемых модулями признаков. Третий модуль представляет из себя два полносвязных слоя. Первый слой третьего модуля обобщает информацию из предыдущих слоев. Второй слой третьего модуля используется для вывода результата работы сети.

3.1 MNIST-digits

Сеть с указанной в данной главе создана и обучена. Данные собраны и представлены в сводной таблице 9.

Таблица 9 – Сводная таблица сравнения разделения данных для MNIST-digits

Сеть	Точность	Время обучения (с)	Время ответа (мс) ²	Высота изображения (пиксел)	Ширина изображения (пиксел)
С полными данными	0,99	92,94	0,62	28	28
Обрезанная по горизонтали	0,98	84,37	0,61	14	28
Обрезанная по вертикали и горизонтали	0,98	84,01	0,60	28	14

Продолжение таблицы 9

Обрезанная по вертикали и горизонтали	0,97	80,57	0,60	14	14
Сильная обрезка	0,81	79,16	0,61	7	7

Проведено сравнение и выявлены закономерности. При уменьшении размера изображения для классификации понижается точность работы нейронной сети, уменьшение времени обучения нейронной сети и сокращение времени ответа.

3.2 MNIST-fashion

По аналогии с предыдущим разделом результаты сверяются с усложненным набором данных — MNIST-fashion. Результаты проведенного эксперимента приведены в сводной таблице 10.

Таблица 10 – Сводная таблица сравнения разделения данных для MNIST-fashion

Сеть	Точность	Время обучения (с)	Время ответа (мс) ²	Высота изображения (пиксел)	Ширина изображения (пиксел)
С полными данными	0,91	85,53	0,58	28	28
Обрезанная по горизонтали	0,87	82,85	0,61	14	28
Обрезанная по вертикали и горизонтали	0,84	82,86	0,60	28	14

Продолжение таблицы 10

Обрезанная по вертикали и горизонтали	0,82	81,56	0,60	14	14
Сильная обрезка	0,61	78,42	0,60	7	7

Далее составлены таблицы с перекрестным сравнением результатов работы полученных в ходе эксперимента нейронных сетей. Сравнения представлены в виде таблиц 11-14.

Таблица 11 – Сравнение точности между сетями

Сеть	С полными данными	Обрезанная по горизонтали	Обрезанная по вертикали и горизонтали	Обрезанная по вертикали и горизонтали	Сильная обрезка
С полными данными	###	0,07	0,07	0,09	0,30
Обрезанная по горизонтали	-0,04	###	0,03	0,06	0,26
Обрезанная по вертикали и горизонтали	-0,07	-0,03	###	0,02	0,23
Обрезанная по вертикали и горизонтали	-0,09	-0,06	-0,02	###	0,21

Продолжение таблицы 11

Сильная об- резка	-0,30	-0,26	-0,23	-0,21	###
----------------------	-------	-------	-------	-------	-----

При уменьшении обрабатываемого изображения наблюдается уменьшение точности нейронной сети. Таким образом эти показатели зависят напрямую. Текущие результаты согласуются с результатами исследования из прошлого раздела текущей работы.

Таблица 12 – Сравнение скорости ответа между сетями

Сеть	С полными данными	Обрезанная по горизонтали	Обрезанная по вертикали и горизонтали	Обрезанная по вертикали и горизонтали	Сильная обрезка
С полными данными	###	-0,03	-0,02	-0,02	-0,02
Обрезанная по горизонтали	0,03	###	0,01	0,01	0,01
Обрезанная по вертикали и горизонтали	0,02	-0,01	###	0,01	0,00
Обрезанная по вертикали и горизонтали	0,02	-0,01	0,00	###	0,00
Сильная обрезка	0,02	-0,01	0,00	0,00	###

Изменение скорости работы сетей крайне малы и вероятно является погрешностью, либо требует изучения в следующих работах.

Таблица 13 – Сравнение времени обучения ответа между сетями

Сеть	С полными данными	Обрезанная по горизонтали	Обрезанная по вертикали и горизонтали	Обрезанная по вертикали и горизонтали	Сильная обрезка
С полными данными	###	2,67	2,66	3,96	7,11
Обрезанная по горизонтали	-2,67	###	-0,01	1,29	4,44
Обрезанная по вертикали и горизонтали	-2,66	0,01	###	1,29	4,44
Обрезанная по вертикали и горизонтали	-3,96	-1,29	-1,30	###	3,14
Сильная обрезка	-7,11	-4,44	-4,45	-3,14	###

Время обучения нейронных сетей снижается с уменьшением размера изображения, подаваемого на вход нейронной сети. За счет уменьшения изображения уменьшается и число необходимых параметров. В следствии происходит ускорение обучения.

3.3 Проверка разделения данных для сверточных сетей.

Вывод

Уменьшение размера изображения, подаваемого на вход нейронной сети, приводит к сокращению времени обучения и снижению точности. При

сравнении скорости ответа сверточных нейронных сетей наблюдаются малые изменения показателя. Возможно, причина заключается во времени реакции на сигнал остановки таймера. Предполагается продолжение изучения скорости ответа нейронных сетей в последующих работах.

4 Параллелизм

Параллелизм возможен как внутри нейронной сети, так и снаружи. Для определения параллелизма на уровне нейронных сетей, в текущем разделе работы, используется понятие внесетевого или внешнего параллелизма. Если, внутри сети используются взаимно параллельные слои — это внутрисетевой или внутренний параллелизм.

4.1 Внесетевой параллелизм

Внешний параллелизм — использование нескольких одновременно запущенных и работающих нейронных сетей. Под внешний параллелизм попадает и использование одного компьютера с несколькими вычислителями. Использование этого подхода сокращает время обучения нейронной сети. Возможны ситуации нехватки оперативной памяти ускорителя при работе с нейросетью. Использование внешнего параллелизма позволяет решить эту проблему за счет объединения ресурсов нескольких вычислителей. Так, в 2012 году этот подход использовался для обучения AlexNet.

Внешний параллелизм возможно подразделить на связный и независимый. При независимом параллелизме сети объединяются глобальной задачей. Каждая из сетей обучается для решения участка задачи. При связном обучении сети тренируются как одна сеть. Алгоритм обучения оказывает влияние на параллельные сети. Результат работы обучения с использованием зависимого параллелизма — распределенная сеть. Результат работы независимого параллелизма — некоторое множество сетей.

4.1.1 Ансамблевое обучение

Ансамблевое обучение нейронных сетей — это техника машинного обучения, которая использует несколько обученных алгоритмов с целью получения лучшей предсказательной эффективности, чем можно было бы получить от каждого алгоритма по отдельности.

Принцип работы ансамбля заключается в следующем: каждая нейронная сеть обрабатывает входные данные независимо и выдаёт свой собственный прогноз. Затем путём агрегации результатов всех нейронных сетей полу-

чается конечный прогноз ансамбля. Возможно применение ансамблевого обучения для построения со зависимых нейронных сетей. В таком случае, обучение происходит сходным, обучению одной сети образом. Также возможно обучение нейронных сетей на нескольких компьютерах. Но при таком обучении разработчику необходимо организовать обучение на нескольких компьютерах:

- реализовать взаимодействие на уровне компьютерных сетей,
- синхронизация процесса обучения на нескольких компьютерах,
- передача и получение параметров обучения,
- передача и получение результатов процесса обучения,
- передача и получение корректировки,
- применение полученной корректировки.

При обучении ансамбля происходит одновременное обучение нескольких нейронных сетей. Корректировка ошибки, в таком случае, происходит аналогично корректировке ошибки единой сети.

В библиотеках машинного обучения уже решена задача связанного внешнего параллелизма и называется ансамблирование [10]. При использовании ансамблирования создается ансамбль из нейронных сетей. Ансамбль — некоторое количество независимо обученных нейронных сетей. При помощи ансамбля возможно строить сложную логику взаимодействия ансамблевых сетей. Возможно составить ансамбль консультантов. В этом случае каждая сеть-консультант решает одну и ту же задачу. Итоговый результат работы ансамбля консультантов — вектор ответов консультантов. Вектор необходимо обработать для получения итогового результата. Не все ансамбли состоят из сетей-консультантов. Возможно создание ансамбля, который решает несколькими сетями одну задачу.

Ансамблирование обладает характеристиками:

- разнообразие моделей — в ансамбль включены различные нейронные сети, выполняющие части глобальной задачи,

- высокая цена подготовки нейронных сетей — ансамбль требует затрачивания большего количества ресурсов на тренировку сети, чем обучение единственной сети,

- предотвращение переобучения нейронных сетей — использование нескольких нейронных сетей, снижает вероятность возникновения проблемы переобучения ансамбля.

Применение ансамблирования зависит от реализуемой задачи. Возможности внесетевого параллелизма обширны, но зависят от решаемых задач.

4.1.2 Слияние нейронных сетей

Существует обратный ход — объединение нескольких нейронных сетей в одну [11]. Объединение применяется для создания мультимодальных нейронных сетей [12]. За счет мультимодальности нейронные сети способны обрабатывать различные формы входной информации: текстовую, аудио, видео, изображений и другие.

Данный подход повысит точность решения задачи, так как объединение нескольких сетей, решающих более конкретные задачи на том же множестве объектов при объединении, будут предоставлять большее количество признаков, и, как следствие повысит точность решения задачи, работая совместно.

4.1.3 Выбор подхода

Использование подхода зависит от того какие конкретно цели преследует разработчик. В зависимости от необходимости использования выбирается алгоритм работы. Возможно также и не использовать ансамблирование и реализовать это самостоятельно, но данный подход не всегда является удобным так как необходимо предусмотреть ожидания ответов от сетей, аккумулировать полученные значения, использовать обработку полученных значений и вывод итогового результата.

Возможный алгоритм реализации ансамблирования сети на этапе исполнения представлен на рисунке 9.

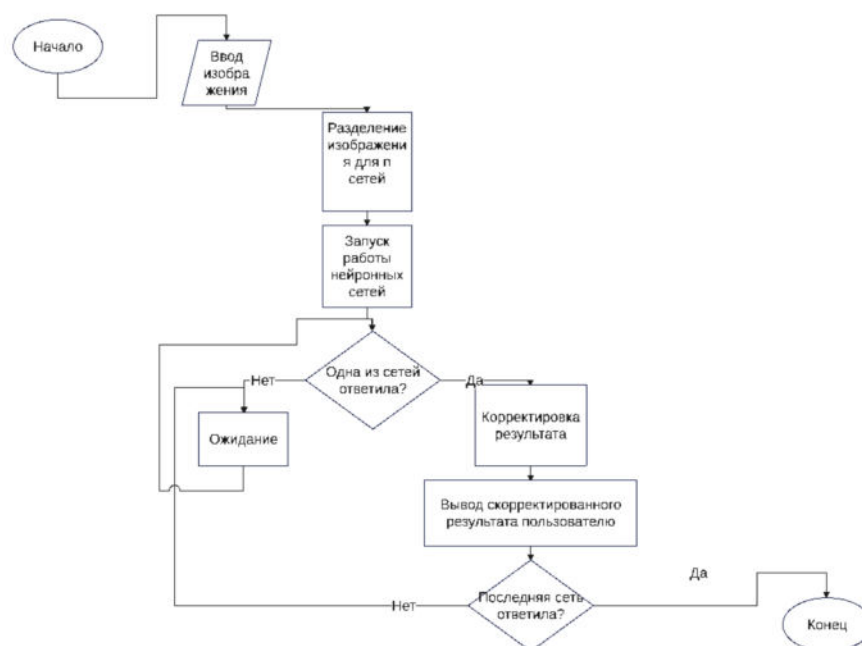


Рисунок 9 – Концептуальный алгоритм ожидания ответа от сетей

Ансамблирование позволяет работать с несколькими сетями снижая нагрузку на вычислитель. Каждая из сетей в ансамбле может обладать различными назначениями. Следовательно, возможно построить сложный механизм взаимодействия нейронных сетей, в котором будут решаться различные задачи, каждой нейронной сетью. Возможно выстроить сложную структуру из нейронных сетей.

Существует возможно комбинирования подходов. Мультимодальность позволит итоговой структуре обрабатывать различные типы пользовательского ввода. Внешняя параллельность позволит распределить сеть и производить манипуляции с данными одновременно. Объединение сетей возможно применить близко к выходным слоям для сложной оценки итоговых значений.

4.2 Внутрисетевой параллелизм

Внутрисетевой параллелизм — явление, при котором, слои внутри нейронной сети образуют более сложную, нелинейную структуру. Применение добавляет измерение в структуре сети — вширь. Появление еще одного измерения делает сеть многомерной.

Внутрисетевой параллелизм позволяет изменить обработку данных нейронной сетью, создавать потоки данных. Этот вид параллелизма был использован в нейронных сетях для классификации изображений [13]. Применение заключалось в добавлении параллельных слоев для извлечения признаков из изображения. Параллельные сверточные слои создавали потоки данных, применяя отличные друг от друга ядра свертки. После потоки данных объединялись в матрицу, которая отправлялась на дальнейшую обработку и классификацию.

Внутрисетевой параллелизм подразделяется на однородный и неоднородный. В однородном параллелизме, слои, располагаемые параллельно, или по типу и установленным параметрам слоя нейронной сети. В неоднородном параллелизме используют различные типы слоев нейронных сетей. Неоднородный параллелизм предоставляет разнообразие слоев, изменяющих данные разными способами. При использовании однородного параллелизма ожидаются одинаковые ответы параллельных слоев. Так как параметры слоев изменяют результат обработки информации слоем.

4.2.1 Неоднородный параллелизм

Неоднородный параллелизм встречается более часто. Для этого есть ряд причин:

- повышение точности полученной сети (за счет разнообразия слоев и разнообразия получаемых ими результатов),
- повышение нелинейности полученной сети (разнообразие позволяет),
- возможность использования нескольких сетей, как одной.

Исходя из описанных причин, в случаях, где возможна техническая возможность обучения одной нейронной сети, можно обойтись без соединения различных сетей в одну или использовать ансамблирование. Таким образом, возможно уменьшить сложности при разработке и эксплуатации сети. Присутствует упрощения размещения объединенной сети. Веса нейронов объединенной нейронной сети возможно поместить в один файл на жестком диске.

Один из примеров, использующих объединение нейронных сетей — GoogLeNet. Архитектура данной сети представлена на рисунке 10.

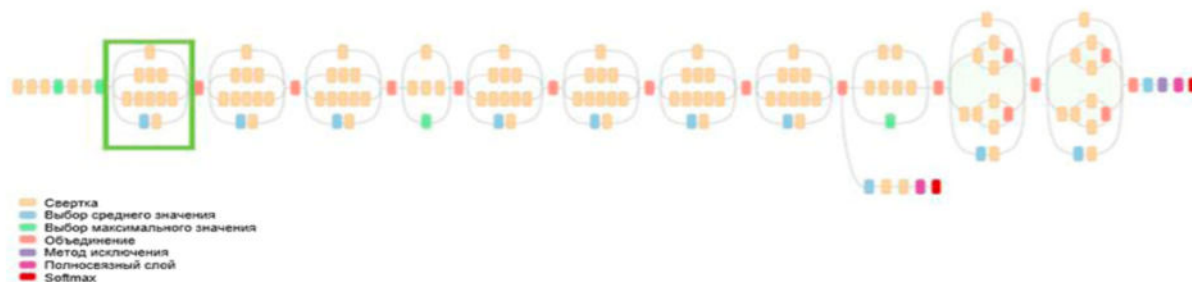


Рисунок 10 – Архитектура GoogLeNet

Особенность данной сети — использование Inception модулей. Inception модуль создан компанией Google и состоит из нескольких параллельных неоднородных сверточных слоев. Модули подключаются последовательно на большей части нейронной сети GoogLeNet. Архитектура данного модуля представлена на рисунке 11.

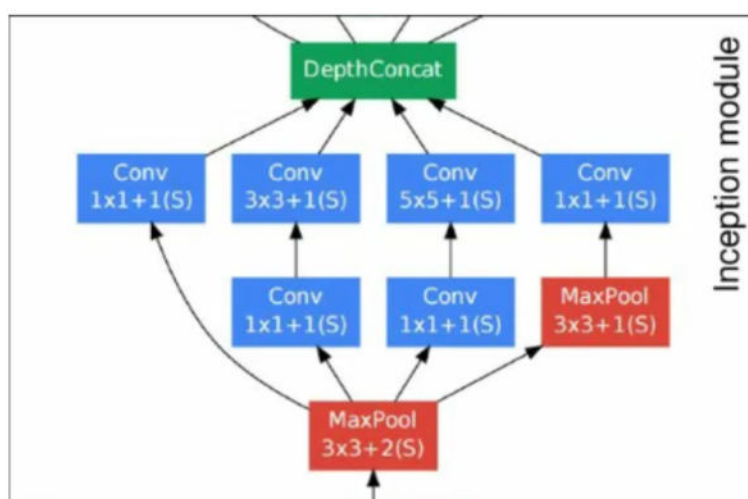


Рисунок 11 – Архитектура модуля Inception

У данного модуля имеются особенности, к которым относятся [14]:

- многоуровневое извлечение признаков (к входным данным применяются несколько фильтров разных размеров и это позволяет сети извлекать информацию разных масштабов и сложностей),
- сокращение размерности (при использовании ядра свертки размером в 1 пиксель, сокращается размер изображения),

- конкатенация (выходы всех фильтров соединяются в одну матрицу и передаются на следующий слой. Таким образом результаты произведенных операций не будут потеряны).

Исходя из вышесказанного возможно выделить преимущества, предоставляемые данным подходом:

- использование Inception модуля снижает необходимое количество параметров, которые использовались бы при отсутствии данного модуля,
- сложность и глубина архитектуры позволяет снизить переобучение, особенно при сочетании с другими методами регулирования переобучения,
- снижение необходимого количества ресурсов приводит к меньшему времени, затрачиваемого для обучения и как следствии — большей эффективности использования ресурсов.

Выбор модулей или слоев в нейронных сетях зависит от типа и условий решаемой задачи. Использование неоднородного параллелизма эффективно. С момента появления неоднородного параллелизма видно активное его использование во многих сетях, в частности: AlexNet, GoogLeNet, YOLO.

4.2.2 Однородный параллелизм

При применении однородного параллелизма точность не должна возрастать. Однако, на практике наблюдалось иное. Нейронная сеть с однородными параллельными слоями демонстрировала большую точность, чем эта же сеть без этих слоев. Принято решение исследовать это явление.

В качестве базовой нейронной сети выбрана сеть LeNet-5. Обучение нейронной сети происходило с использованием графического процессора AMD Radeon RX 6900 XT. Для построения и обучения использовалась технология PyTorch. Набором обучения выбран MNIST-fashion.

LeNet-5 модифицирована для добавления параллельных слоев. Создание сети представлено на рисунке 12.

```

class LeNet5(nn.Module):
    def __init__(self, num_classes, par_nets_count):

        self.cn_layers = [self.create_cnn_layer()] * par_nets_count
        self.fc = nn.Linear(256 * par_nets_count, 120 * par_nets_count)
        self.relu = nn.ReLU()
        self.fc1 = nn.Linear(120 * par_nets_count, 84 * par_nets_count)
        self.relu1 = nn.ReLU()
        self.fc2 = nn.Linear(84 * par_nets_count, num_classes)
        pass

    @staticmethod
    def create_cnn_layer():
        return nn.Sequential(
            nn.Conv2d(1, 6, kernel_size=5, stride=1, padding=0),
            nn.BatchNorm2d(6),
            nn.ReLU(),
            nn.MaxPool2d(kernel_size = 2, stride = 2),
            nn.Conv2d(6, 16, kernel_size=5, stride=1, padding=0),
            nn.BatchNorm2d(16),
            nn.ReLU(),
            nn.MaxPool2d(kernel_size = 2, stride = 2)
        ).to(device)

    def forward(self, x):
        p_data = []
        for layer in self.cn_layers:
            p_data.append(layer(x))
            pass

        out = torch.concat(p_data, -1)
        out = out.reshape(out.size(0), -1)
        out = self.fc(out)
        out = self.relu(out)
        out = self.fc1(out)
        out = self.relu1(out)
        out = self.fc2(out)
        return out

```

Рисунок 12 – Архитектура модифицированной LeNet-5

Изображение одновременно поступает на каждую из параллельных слоев. Результаты работы каждого параллельного слоя объединяются в общую матрицу. Необходимо изменить число нейронов в полносвязных слоях. Это обусловлено использованием конкатенации матриц, выходящих из сверточных сетей. При использовании среднего значения вместо объединения данных

из параллельных слоев возможно потерять точность, но эксперимент с применением усреднения значений матриц реализован.

Составлена схематичная структура сети. Данная структура представлена на рисунке 13.



Рисунок 13 – Схематичная архитектура модифицированной LeNet-5

Сравнение происходит по количеству слоев. Для проведения сравнения выбрано сети с:

- отсутствием параллельных слоев,
- тремя слоями,
- пятью слоями,
- семью слоями.

Полученные данные представлены в таблице

Таблица 14 – Результаты сравнения параллельных однородных сетей

Сеть	Точность	Время обучения (с)	Время отклика (мс)
Без параллельности	0,829551333	0,624403	87,24154133
3 параллельных слоя	0,834759667	0,666079667	105,917876

Продолжение таблицы 14

5 параллельных слоев	0,842058	0,696461	128,245273
7 параллельных слоев	0,844214333	0,734734667	145,7031113

Наблюдается увеличение точности при увеличении количества параллельных сетей. Аналогично происходит с показателями: время обучения, время отклика сети. Изменения точности нелинейное и приближается к некоторому пределу. Предел изменений планируется исследовать в последующих работах.

Во время обучения собраны метрики точности для каждой эпохи. Визуализация данных представлена на рисунке 14.

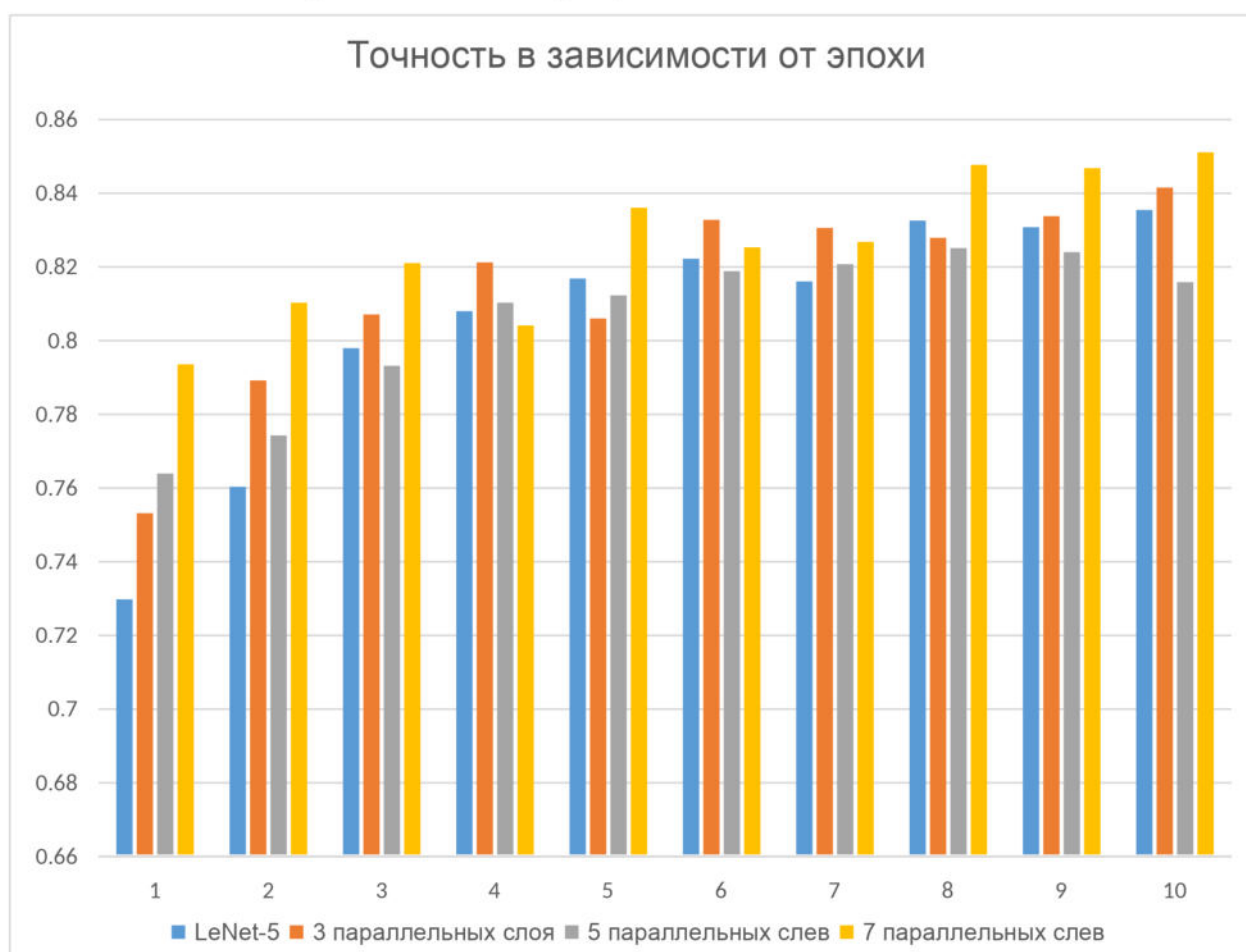


Рисунок 14 – Изменение точности в зависимости от эпохи

При сравнении данных визуализации наблюдается зависимость между количеством параллельных слоев и точности в рамках эпохи. Эта зависимость является сложной, так как, в ряде случаев сеть с меньшим количеством параллельных слоев демонстрирует результат лучше, чем сеть с большим количеством.

В следующем эксперименте конкатенация заменена на поэлементное усреднение результатов параллельных сетей. Изменённая архитектура сети представлена на рисунке 15.

```
class LeNet5Means(nn.Module):
    def __init__(self, num_classes, par_nets_count):

        self.cn_layers = [self.create_cnn_layer()]
        self.fc = nn.Linear(256, 120)
        self.relu = nn.ReLU()
        self.fc1 = nn.Linear(120, 84)
        self.relu1 = nn.ReLU()
        self.fc2 = nn.Linear(84, num_classes)
        pass

    @staticmethod
    def create_cnn_layer():
        return nn.Sequential(
            nn.Conv2d(1, 6, kernel_size=5, stride=1, padding=0),
            nn.BatchNorm2d(6),
            nn.ReLU(),
            nn.MaxPool2d(kernel_size = 2, stride = 2),
            nn.Conv2d(6, 16, kernel_size=5, stride=1, padding=0),
            nn.BatchNorm2d(16),
            nn.ReLU(),
            nn.MaxPool2d(kernel_size = 2, stride = 2)
        ).to(device)

    def forward(self, x):
        p_data = []
        for layer in self.cn_layers:
            p_data.append(layer(x))
            pass

        out = torch.mean(torch.stack(p_data), dim=0)
        out = out.reshape(out.size(0), -1)
        out = self.fc(out)
        out = self.relu(out)
        out = self.fc1(out)
        out = self.relu1(out)
        out = self.fc2(out)
        return out
```

Рисунок 15 – Архитектура модифицированной LeNet-5 используя усреднение

Произведен сбор и анализ метрик полученных сетей. Результаты представлены в таблице 15.

Таблица 15 – Результаты сравнения параллельных однородных сетей с усреднением

Сеть	Точность	Время обучения (с)	Время отклика (мс)
Без параллельности	0,829551333	0,624403	87,24154133
3 параллельных слоя	0,830480333	0,61663	87,03397167
5 параллельных слоев	0,831608333	0,620447	87,57953433
7 параллельных слоев	0,827461667	0,601139	83,985882

Наблюдается зависимость между точностью и количеством параллельных слоев внутри сети. Выявленная зависимость нелинейна и стремится к некоторому пределу. Этот предел планируется исследовать в последующих работах. Наблюдается обратная, нелинейная зависимость между количеством параллельных слоев и временем обучения. Также, наблюдается обратная, нелинейная зависимость между количеством параллельных слоев и временем отклика сети.

Собраны данные изменения точности в пределах эпохи. Визуализированны на рисунке 16

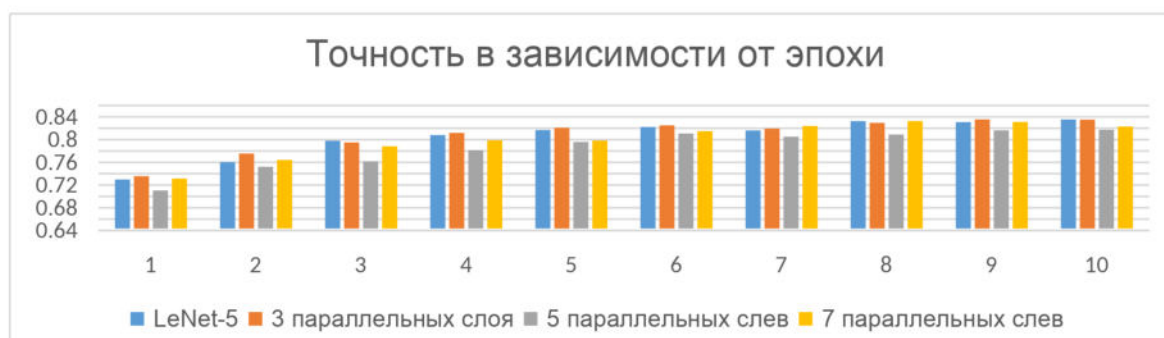


Рисунок 16 – Точность в зависимости от эпохи для усредненной сети

В полученной визуализации наблюдается близкие точности между оцениваемыми сетями в рамках эпохи. Сеть с семью параллельными слоями не показывает лучший результат в большинстве случаев.

Устройства, на которых производится обучение, потенциально могут оказывать влияние на обучение. С целью исключения потенциального влияния решено провести подтверждение результатов эксперимента, в котором использовалась конкатенация матриц на различных устройствах. Во время обучения начальное число генератора случайных значений фиксировалось.

Обучение производилось с использованием: центрального процессора AMD Ryzen 5 5600X, графической карте AMD Radeon 6900XT, центральном процессоре Apple M1. В качестве тестируемой сети выбрана сеть с 7 параллельными однородным слоями.

Произведено сравнение обучения между AMD Ryzen 5 5600X и AMD Radeon 6900XT. По собранным данным построена визуализация, представленная на рисунке 17.

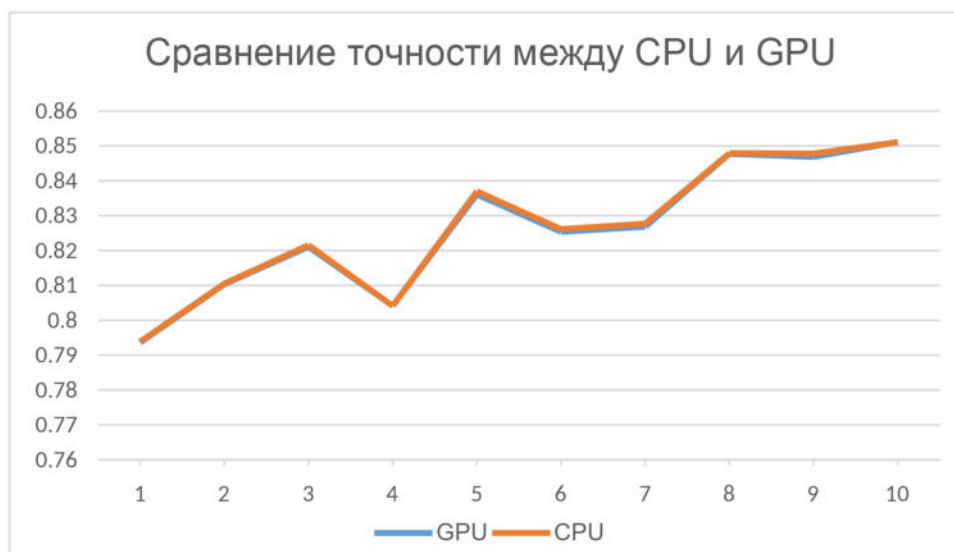


Рисунок 17 – Сравнение точности с использованием центрального процессора и графического процессора

По оси абсцисс представлен номер эпохи. По оси ординат представлена точность нейронной сети. При использовании чипов AMD разница не наблюдается.

Произведено сравнение обучения между AMD Radeon 6900XT и Apple M1. По собранным данным построена визуализация, представленная на рисунке 18.

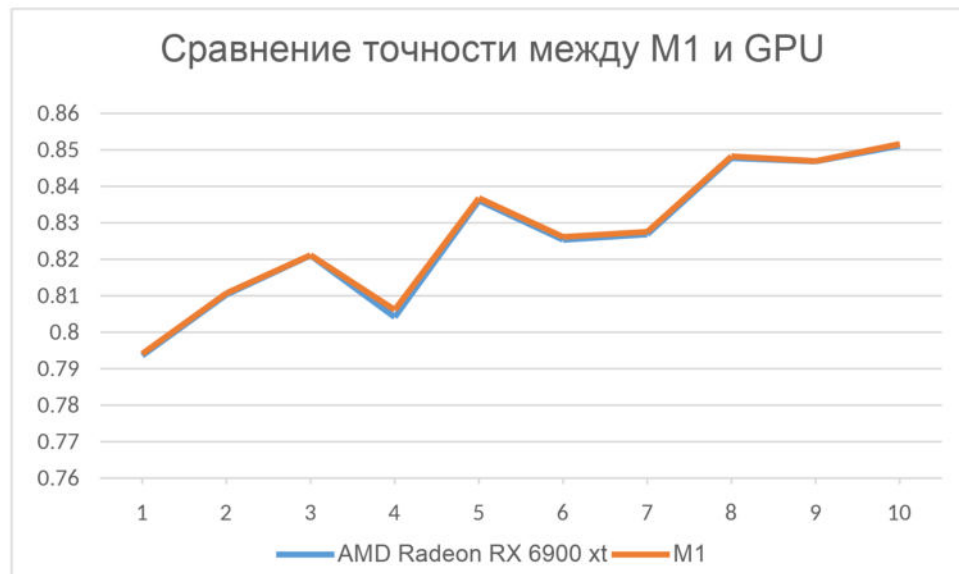


Рисунок 18 – Сравнение точности между M1 и GPU

Присутствует малое отличие на 4 эпохе. Возможной причиной являются операции с плавающей точкой. При округлении до 4 знака после запятой различия отсутствуют. Возможно данное поведение обосновывается ошибкой при выполнении операций с плавающей точкой.

Так как фиксация генератора случайных значений демонстрирует неизменные результаты на различных устройствах, необходимо произвести сравнение процесса обучения с различными начальными ключами данного генератора. Произведено обучение сети с семью параллельными слоями. Собраны и проанализированы данные. Результаты представлены на рисунке 19.

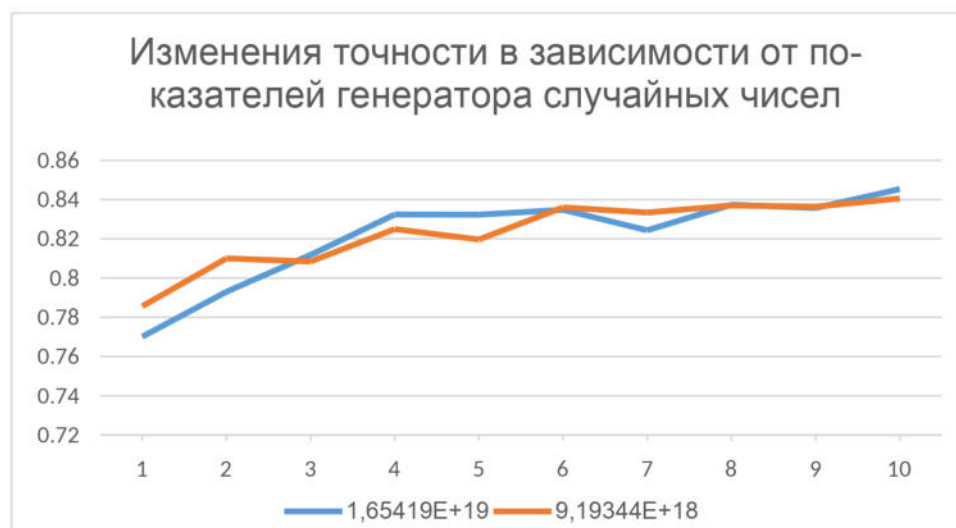


Рисунок 19 – Изменение показателей точности для различных показателей генератора случайных значений

При отличном начальном значении генератора случайных чисел происходят изменения в процессе обучения сетей. Точность полученных сетей отличается и кривые практически не накладываются друг на друга на графике. Следовательно, основное влияние оказывает работа генератора случайных значений.

Генератор случайных значений используется в нейронной сети для инициализации весов нейронов в нейронной сети случайными числами [15]. Следовательно, во время инициализации весов нейронов сети, однородные параллельные слои, получают отличающиеся друг от друга начальные веса. Это приводит к приближению, либо отдалению веса нейрона с одинаковым индексом в разных параллельных слоях от оптимального значения. Следовательно, результат работы каждой параллельной сети может отличаться.

При использовании конкатенирующего слоя стабильно наблюдается увеличение точности. Следовательно, повышается вероятность увеличения точности при использовании однородных параллельных слоев в нейронной сети. Для каждого элемента конкатенированной матрицы существует нейрон в полносвязной сети. За счет регуляции весов производится изменение влияние элементов из этой матрицы на сеть. Следовательно, оптимизатор сети, за счет минимизации функции потерь нейронной сети, изменит веса в большую сторону для более подходящих элементов общей матрицы.

Усреднение приводит к невозможности воспользоваться оценкой конкретных элементов независимо. Так как, отсутствует возможность учитывать точность параллельных сетей. Итоговая точность нестабильна при повторях эксперимента. При этом на большинстве эпох однородные параллельные сети с усреднением данных показывают себя лучше, чем оригинальная сеть.

Возможно внедрить систему оценки. После обучения, происходит подключение комбинаций параллельных сетей и тестирование итоговой сети. После получения и анализа результатов, возможно отключить часть параллельных сетей. Следовательно, в варианте усреднения результата параллельных однородных сетей, возможна последующая оптимизация полученной нейронной сети. Исключение слоев, демонстрирующих, худший результат, в сравнении с другими слоями, упрощается за счет неизменности размера вектора, поступающего на полносвязные сети. Следовательно, используя усреднение матриц, поступающих из параллельных слоев, возможно добавление, либо исключение параллельных слоев.

При использовании ансамблирования, слои, отвечающие за отбор признаков, возможно вынести в отдельную нейронную сеть. Также, за счет применения ансамблирования возможно перевести внутрисетевой параллелизм во внешний параллелизм. Этот подход позволяет еще более тонко настроить сбор метрик и выбор сети для дальнейшего использования. Но важно учитывать повышение потребления ресурсов при использовании ансамблирования. Возможно и использование ансамблирования только лишь для оценки. После проведения оценки, полученные сети возможно объединить.

4.3 Варианты использования

В проведенных экспериментах, наблюдается зависимость между количеством слоев и точностью итоговой нейронной сети. Зависимость существует как при конкатенации матриц, выходящих с однородных параллельных слоев, так и при использовании усреднения соответствующих значений в матрицах. Также, в случае использования конкатенации, при увеличении количе-

ства однородных параллельных слоев возрастает время ответа нейронной сети и время, необходимое, для ее обучения.

Повышение точности происходит, за счет различной инициализации начальных весов у нейронов в параллельных слоях. Следовательно, при конкатенации результатов работы параллельных слоев, возможно увеличить точность. Увеличение точности произойдет за счет установления большего веса нейрону, взаимодействующему, с более точно подобранным параметром из параллельных сетей.

Возможно использование внешнего параллелизма при необходимости обучения нейронной сети несколькими вычислителями в ситуации, когда это невозможно одним вычислителем, по показателям общего времени обучения или техническими ограничениями ускорителя.

Еще одним вариантом использования является возможность производить перекрестную оценку сетей, участвующих в обучении. Таким образом за одно обучение возможно подготовить несколько сетей, произвести их оценку взаимодействия друг с другом, и провести выборку наиболее подходящей сети или сетей. За счет использования перекрестной оценки возможно в рамках одного обучения произвести более быструю подготовку итоговой сети. Эта возможность обусловлена отсутствием замедления обучения нейронной сети кратно количеству параллельных слоев.

Важно применять представленные подходы исходя из условий и особенностей решения задачи. Сами условия происходят из выбора данных, с которыми происходит работа. Таким образом, возможно построить несколько гипотез о взаимодействии созданных сетей между собой, и проверить их в рамках обучающего процесса. При такой проверке возможно собирать метрики и отслеживать процесс обучения различных сетей в режиме реального времени, и еще на этапе обучения, произвести исключение сети, которая по мнению исследователя менее подходит для представленных данных. Но при таких действиях придется произвести остановку обучения сетей, исключить сеть и запустить обучение снова. Также, возможно предусмотреть необходимость оста-

новки при реализации обучения нейронной сети и добавить возможность исключать эту сеть из обучения. В таком случае, возможно, потребуются вмешательство человека, либо разработки алгоритма, автоматически исключающего неподходящие сети. Данное решение повышает сложность разработки.

Использование внутрисетевого параллелизма предлагает немало возможностей. Возможность использования различных параллельных слоев позволяет сильно изменить саму нейронную сеть. Использование параллельных слоев позволяет разделить систему по сложности. Также, позволяет добиться сложного, нелинейного поведения в рамках поставленной задачи. Данный подход может применяться не только в ситуации, когда необходимо получить большой набор признаков с входных слоев, но и при необходимости, использовать его для обработки полученных признаков. При применении параллельности, возможно создавать крайне сложные и многообразные структуры похожие на деревья или графы. При этом, возможно использовать и различные выходы из сети. Таким образом, классифицирующая сеть может применить данный подход для сложной классификации. Например, сеть будет иметь пять выходов. Попадание на конкретный выход результата зависит от размеров объекта или размера класса, к которому принадлежит объект. Например, большие объекты проходя классификацию, в слоях, предназначенных для классификации таких объектов, могут не передавать данные об этом объекте на последующие сети и произвести вывод результата сразу. Аналогично и в ситуации, когда классифицируемый объект малых размеров. Подобный подход встречается в классификаторах, например YOLO [16]. Архитектора v4 представлена на рисунке 20.

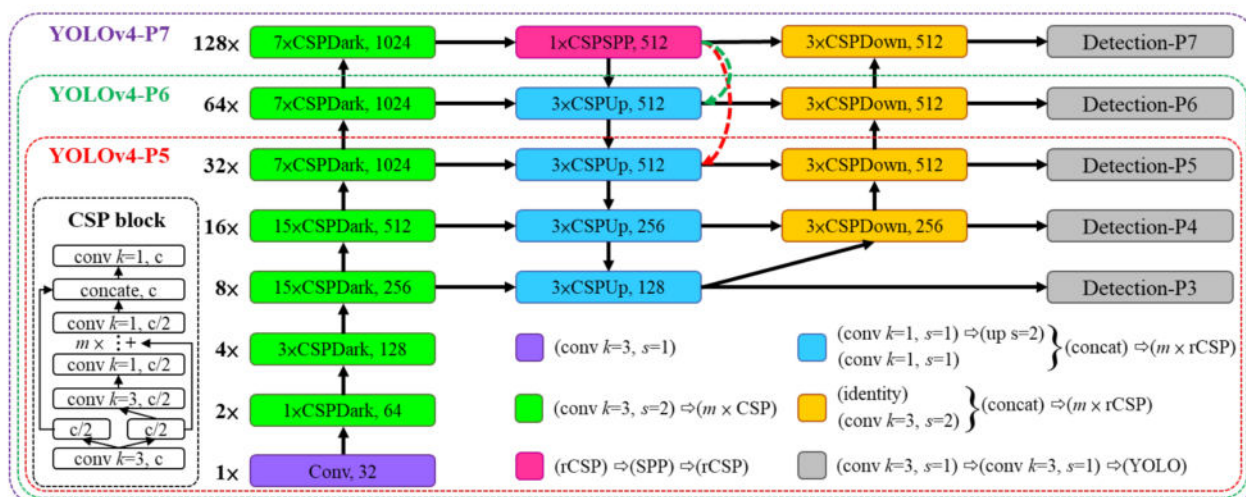


Рисунок 20 – Архитектура YOLOv4

Как видно из данного рисунка после получения изображения, оно проходит через несколько сверточных слоев, где после этого, направляется на следующие, аналогичные слои в одном направлении, и на классификаторы в другом.

При этом замеры работы YOLOv4 исследователями указывают на возможность проведения детектирования не только изображений, но и видео. YOLOv4 может работать в режиме точности с малым показателем кадров в секунду, либо с большим количеством кадров в секунду, но меньшей точности. Важным является как количество кадров с секунду, так и точность. При малом количестве кадров в секунду не получится своевременно реагировать на изменения на видео. Однако при низкой точности теряется смысл использования классификатора. Сравнение классификаторов показано на рисунке 21

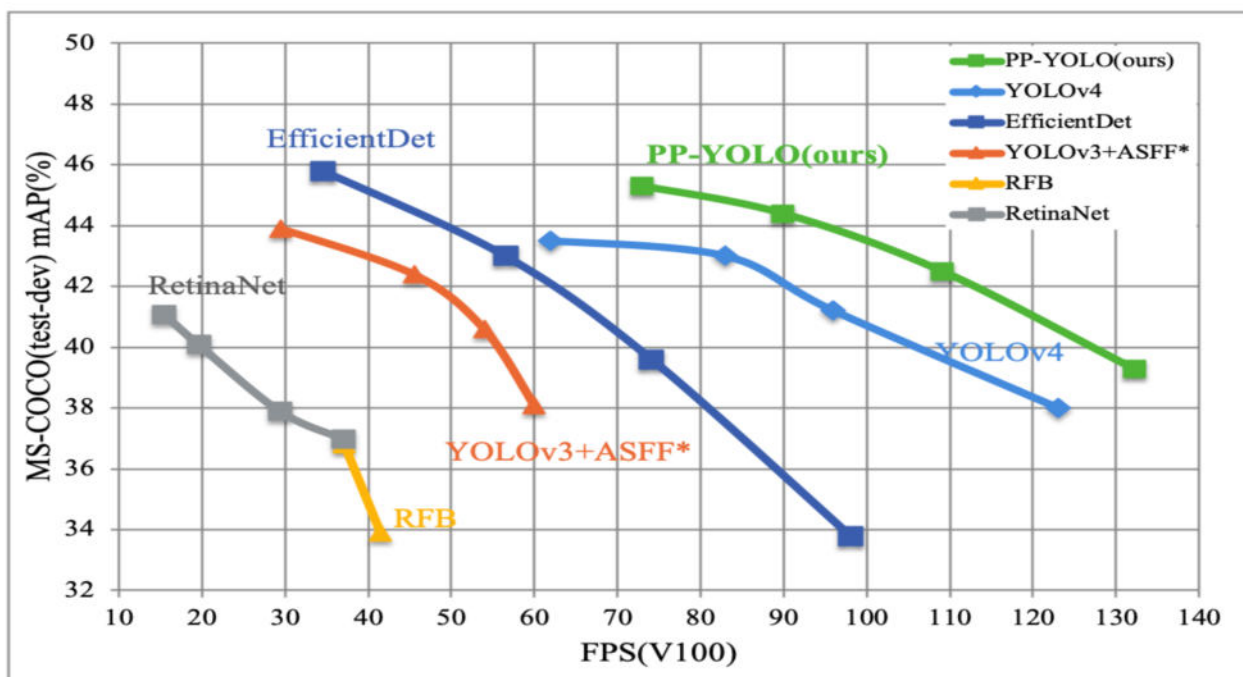


Рисунок 21 – Сравнение детекторов

Как видно из рисунка, точность в 50% не была достигнута ни одной представленной сетью. При этом YOLOv4 позволяет работать с видео с частотой свыше 60 кадров в секунду.

В данный момент все крупные сети используют внутренний параллелизм. И причиной этого возросшая сложность решаемых задач. Задача оцифровки рукописных знаков давно решена. И в данный момент поставлены и активно решаются сложные задачи, такие как управление беспилотным транспортом при помощи нейронных сетей. В рамках этой задачи критически важны точность и количество обрабатываемых кадров, устройствами наблюдения. В рамках нескольких кадров может произойти потенциально опасная ситуация, на которую необходимо незамедлительно отреагировать. Параллельно происходит оптимизация в аппаратном обеспечении. Это необходимо для увеличения скорости работы нейронных сетей, а также для возможностей параллельной работы нескольких сетей. Еще оптимизация требуется в направлении энергоэффективности аппаратных составляющих.

Чаще встречаются сеть со сложной, древовидной структурой. Использование этой структуры позволяет решать сложные задачи. Чем выше показатель нелинейности нейронной сети, тем более сложные задачи она может ре-

шать. Многие простые сети не способны обобщить данные даже для относительно простых наборов данных. Это прослеживается в ходе выполнения текущей работы, где часть обозреваемых нейронных сетей не обладают 100% точностью на наборе данных MNIST-fashion.

Примером одной из структурно сложных сетей, занимающихся классификацией, является PNASNet-5 [17]. Фрагмент архитектуры представлен на рисунке 22.

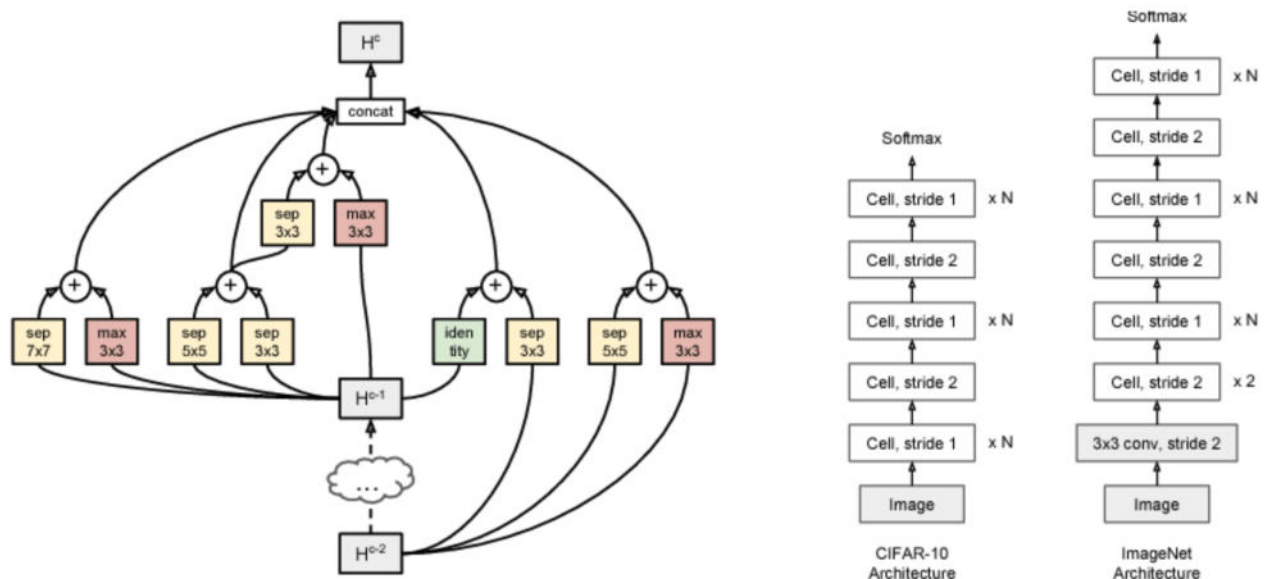


Рисунок 22 – Фрагмент архитектуры PNASNet-5

Как видно, фрагмента, модуль сети имеет большое разветвление и производит большое количество расчетов. Также видно и применение объединения данных.

PNASNet-5 – большая модель, состоящая из схожих по строению модулей. Разветвленность в рамках архитектуры PNASNet-5 не только на уровне модулей. Разветвленной является сама сеть и взаимодействие между модулями нейронной сети. Таким образом достигается высокие показатели точности и возможности запоминания большого количества информации нейронной сетью. Наблюдается тенденция к усложнению архитектур нейронных сетей и повышению количества задач, решаемых в рамках одной сети. Примером являются мультимодальные нейронные сети, например GPTv4o.

ЗАКЛЮЧЕНИЕ

В рамках данной работы произведено изучение предметной области, выделены характеристики слоев и нейронных сетей, поставлены эксперименты, получены и проанализированы данные, а также составлены варианты использования исходя из полученных данных.

При использовании разделения данных важны типы используемых слоев в нейронных сетях. Так, согласно полученным данным, скорость обучения нейронной сети возрастает по мере уменьшения количества размера, передаваемого изображения, но имеет предел. Предел ускорения планируется изучить в последующих работах.

В многомерных, сверточных сетях, отмечается зависимость между размером изображения, обрабатываемого нейронной сетью, и точностью полученной сети, а также между размером изображения и временем обучения нейронной сети. При уменьшении размера изображения происходит уменьшение точности и времени, необходимого для обучения сети. В случае если потеря точности определяется разработчиком как не критичная, возможно воспользоваться уменьшением размера изображения. В этом случае разработчику необходимо определить предел уменьшения размера изображения для предотвращения слишком большой потери точности.

Проведено исследование внесетевого параллелизма. Произведены эксперименты, собраны данные и представлены варианты использования.

Использование внутрисетевого параллелизма зависит от целей разработчика и решаемой задачи. В зависимости от использования выбирается алгоритм работы, набор слоев в сети, данные для обучения и метод обучения нейронной сети. Возможна комбинация внутрисетевого параллелизма и внешнего параллелизма, например ансамблирования. Помимо ансамблирования возможна самостоятельная реализация, в ситуациях, где требуется использовать взаимодействие в рамках разных компьютерных сетях.

Комбинирование внутрисетевого параллелизма с внешним параллелизмом позволяет работать с несколькими сетями, снижая нагрузку на вычисли-

тель. Обосновано существование проблемы пониженной точности при уменьшении размера используемых данных. Но при передаче данных, в которых не происходило модификации размера, возможно построить независимые сети, дополняющие друг друга — сети-консультанты. Помимо сетей-консультантов, возможно использовать различные нейронные сети в связке. За счет комбинирования различных нейронных сетей возможно расширить объем решаемых задач или повысить качество решения поставленной задачи.

Таким образом при использовании ансамблирования возможно решить несколько задач, либо углубленно решить одну задачу, в которой, возможно распределить вычисление признаков не несколько различных нейронных сетей. После этого этапа предоставить полученные признаки на сети, отбирающие признаки, а после, на сети анализирующие признаки, которые, в свою очередь, предоставят результат работы общей системы. Возможно комбинировать внесетевой параллелизм с внутрисетевым параллелизмом и объединением сетей сетей.

Объединенные сети подходят в ситуациях, когда не требуется разделять сети на разные вычислители и требуется одна сеть для решения задачи. Либо при объединении сетей, решающие одну и ту же задачу, для увеличения точности за счет структурных различий в объединяемых сетях. Существуют технологии позволяющие распараллеливать вычисления внутри одного компьютера в рамках одной нейронной сети. Таким образом, возможно назначить для разных слоев в нейронной сети, разные ускорители. Но это возможно только, если есть несколько устройств обработки информации.

При использовании внешнего параллелизма возможно обучать нейронные сети несколькими компьютерами. Эту особенность внешнего параллелизма применяют при высоких требованиях обучаемой нейронной сети к количеству доступных ресурсов. Также это позволяет сократить время на обучение нейронных сетей из-за параллельности в обучении. Этим подходом возможно разделить сложную нейронную сеть на несколько более простых, упростив разработку. Упрощение происходит за счет разработки простых блоков в виде

нейронных сетей. Также упрощение достигается за счет большего контроля полученного результата. Недостатком является повышенное использование ресурсов и, при обучении на нескольких компьютерах, появление задержки в коммуникации между компьютерами. После этого, для использования, сеть возможно произвести объединение обученных сетей в одну.

Отдельным вариантом использования является и возможность работы внешне параллельных сетей как независимые сети и возможность их использования в других сочетаниях для решения иной или той же задачи с применением других техник или сетей для обработки информации. Важным преимуществом данного подхода является и возможность частичной загрузки общей нейронной сети. Это означает загрузку в память сначала одной сети, а после получения результата, выгрузку первой и загрузку следующей сети. Это может повторяться до тех пор, пока не будет получен окончательный результат. Применение возможно в ситуации, где у вычислителя недостаточно памяти для обработки нейронной сети целиком. Таким образом, возможно внедрить сложные, нелинейные вычисления на слабые по мощности устройства и использовать некоторые сложные сети на носимых устройствах, к которым относятся мобильные устройства.

Еще одним вариантом использования является возможность производить перекрестную оценку сетей, участвующих в обучении. Таким образом, за одно обучение возможно подготовить несколько сетей, произвести оценку их взаимного взаимодействия и провести выборку наиболее подходящей сетей. Применение этого подхода представляет возможность строить сложное взаимодействие и оценку полученного результата. Таким образом, возможно построить несколько параллельных слоев в сети и комбинаций их взаимодействия между собой. После или во время обучения проверить их. При такой проверке возможно собирать метрики и отслеживать процесс обучения различных сетей в режиме реального времени и, еще на этапе обучения произвести исключение сетей, с худшим результатом. Возможно создание алгоритма, выполняющего исключение слоев нейронных сетей с худшими показателями.

Показатели, влияющие на исключение, задаются программистом. Создание и использование такого алгоритма повышает сложность разработки.

Использование внутрисетевого параллелизма предлагает немало возможностей. Возможность использования различных параллельных слоев позволяет изменить работу с данными, добавить нелинейность и сложность обработки данных. Использование параллельных слоев позволяет разделить систему по сложности. Также позволяет добиться сложного, нелинейного поведения в рамках поставленной задачи увеличивая: качество ответов нейронной сети, количество возможных применений. Данный подход может применяться не только в ситуации, когда необходимо получить большой набор признаков с входных слоев, но и при необходимости, использовать его для обработки этих признаков. При применении параллельности возможно создавать крайне сложные и многообразные структуры похожие на деревья или графы. При этом возможно использовать и ранний выход из нейронной сети. Таким образом, классифицирующая сеть может применить данный подход для сложной классификации.

Применение однородного внутрисетевого параллелизма позволяет оценить возможное улучшение нейронной сети и минимизировать влияние генератора случайных значений. В случае, где не применяются однородный параллелизм, обучение нейронной сети будет возвращать нейронную сеть с разными показателями точности. При этом разница в точности может быть как значительной, так и малой. Если внутрисетевой применяется, значение точности готовой сети, после обучения отличается в меньшую сторону, в сравнении с сетью без однородных параллельных слоев. Следовательно, существует возможность применения однородного внутрисетевого параллелизма для оценки максимальных возможностей обученной нейронной сети и. В следствии возможно за одно обучение подготовить наиболее оптимальную нейронную сеть.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. История и развитие нейронных сетей // vr-app URL: <https://vr-app.ru/blog/istoriia-i-razvitie-neironnyx-setei/> (дата обращения: 12.05.2024).
2. Порошин И.Е., Титов М.А. Применение графических процессоров с технологией CUDA в высокопроизводительных системах с искусственными нейронными сетями // Молодой учёный. - 2020. - №25 (315). - С. 123-125.
3. Стрелец А.И., Протопопова Ю.Д., Тоичкин Д.В., Ключникова Б.В. АНАЛИЗ ТЕНЗОРНЫХ ПРОЦЕССОРОВ НА ПРЕДМЕТ ЭФФЕКТИВНОСТИ ПРИМЕНЕНИЯ ДЛЯ РАСЧЁТА СИСТОЛИЧЕСКИХ МАССИВОВ НЕЙРОННЫХ СЕТЕЙ // Форум молодых ученых. - 2018. - №7 (23). - С. 914-919.
4. Подборка фреймворков для машинного обучения // Хабр URL: <https://habr.com/ru/companies/parallels/articles/317994/> (дата обращения: 12.05.2024).
5. PyTorch и TensorFlow: отличия и сходства фреймворков // Neurohive URL: <https://neurohive.io/ru/tutorial/pytorch-vs-tensorflow/> (дата обращения: 12.05.2024).
6. Архитектуры нейросетей // Хабр URL: <https://habr.com/ru/companies/nix/articles/430524/> (дата обращения: 12.05.2024).
7. AlexNet — свёрточная нейронная сеть для классификации изображений // Neurohive URL: <https://neurohive.io/ru/vidy-nejrosetej/alexnet-svjortochnaja-nejronnaja-set-dlja-raspoznavanija-izobrazhenij/> (дата обращения: 12.05.2024).
8. Полносвязные слои нейронных сетей в машинном обучении // Хабр URL: <https://habr.com/ru/articles/718044/> (дата обращения: 12.05.2024). Maximum Pooling // Data Science Portfolio URL:

- <https://sourestdeeds.github.io/blog/maximum-pooling/> (дата обращения: 12.05.2024).
9. Лосева Е. Д., Антамошкин А. Н. Алгоритм автоматизированного формирования ансамблей нейронных сетей для решения сложных задач интеллектуального анализа данных // Известия Тульского государственного университета. Технические науки. - 2017. - №4. - С. 234-243.
 10. On Combining Artificial Neural Nets // researchgate URL: https://www.researchgate.net/publication/220233566_On_Combining_Artificial_Neural_Nets (дата обращения: 12.05.2024).
 11. Стефаниди Антон Федорович ИССЛЕДОВАНИЕ МУЛЬТИМОДАЛЬНЫХ АЛГОРИТМОВ БИОМЕТРИЧЕСКОЙ ИДЕНТИФИКАЦИИ НА ОСНОВЕ МЕТОДОВ ЦИФРОВОЙ ОБРАБОТКИ РЕЧЕВЫХ СИГНАЛОВ И ИЗОБРАЖЕНИЙ: дис. канд. Хрящев Владимир Вячеславович Радиотехника, в том числе системы и устройства телевидения наук: 2.2.13. - Ярославль, 2022. - 130 с.
 12. Винокуров И.В. Распознавание цифровых последовательностей с использованием свёрточных нейронных сетей // Программные системы: теория и приложения. - 2023. - №1 (56). - С. 1-34.
 13. Inception Module // deepai URL: <https://deepai.org/machine-learning-glossary-and-terms/inception-module> (дата обращения: 12.05.2024).
 14. Методы инициализации весовых коэффициентов нейронной сети // mql5 URL: https://www.mql5.com/ru/neurobook/index/about_ai/initialization (дата обращения: 12.05.2024).
 15. YOLOv4: высокоскоростное и точное обнаружение объектов // ultralytics URL: <https://docs.ultralytics.com/ru/models/yolov4/> (дата обращения: 12.05.2024).
 16. PNASNet // paperswithcode URL: <https://paperswithcode.com/lib/timm/pnasnet> (дата обращения: 12.05.2024).