МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение

высшего образования

**«КУБАНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»**

**(ФГБОУ ВО «КубГУ»)**

**Факультет компьютерных технологий и прикладной математики**

**Кафедра информационных технологий**

**КУРСОВАЯ РАБОТА**

**МЕТОДЫ РАЗРАБОТКИ И ОБУЧЕНИЮ НЕЙРОННЫХ СЕТЕЙ СРЕДСТВАМИ БИБЛИОТЕКИ KERAS**

Работу выполнил А.Н. Граборов

(подпись)

Направление подготовки 01.03.02 Прикладная математика и информатика

Направленность (профиль) «Системное программирование и компьютерные технологии» (Математическое и программное обеспечение вычислительных машин)

Научный руководитель

доц., канд. физ.-мат. наук В.В. Подколзин

(подпись)

Нормоконтролер

ст. преп. А.В. Харченко

(подпись)

Краснодар

2018

**РЕФЕРАТ**

Курсовая работа содержит 29 страниц, 9 рисунков, 8 источников.

НЕЙРОННЫЕ СЕТИ, РАСПОЗНОВАНИЕ ОБРАЗОВ, ПРОГНОЗИРОВАНИЕ, ИСКУССТВЕННЫЙ ИНТЕЛЕКТ

Объектом исследования являются нейронные сети, их создание и применение в различных задачах.

Цель курсовой работы – изучение работы нейронной сети, выбор наилучшей среды разработки нейронных сетей.

Результатом проделанной работы является знакомство с библиотекой Keras, всеми основными типами нейронных сетей, способов их обучения, подробное изучение работы сверточных нейронных сетей для распознавания образов на изображении, создание небольших программ с использованием нейронных сетей для личных целей.

СОДЕРЖАНИЕ

[Введение 4](#_Toc516320551)

[1 Введение в искусственные нейронные сети 5](#_Toc516320552)

[1.1 История развития ИНС 5](#_Toc516320553)

1.2 Проблемы решаемые в рамках ИНС 6

1.3 Структура искусственного нейрона 7

1.4 [Архитектура нейронной сети 9](#_Toc516320553)

[1.5 Обучение нейронных сетей 14](#_Toc516320553)

[1.5.1 Алгоритм обратного распространения ошибки 1](#_Toc516320554)8

1.6 [Некоторые проблемы ИНС 21](#_Toc516320555)

[2 Создание нейронных сетей 23](#_Toc516320556)

[2.1 Среда разработки искусственных нейронных сетей 24](#_Toc516320557)

[2.2 Пример элементарной сети на Keras 24](#_Toc516320557)

[3 Программная реализация нейронной сети 26](#_Toc516320570)

[3.1 Постановка задачи 26](#_Toc516320571)

[3.2 Выбор среды разработки 26](#_Toc516320572)

[3.3 Описание алгоритма реализации 28](#_Toc516320572)

[Заключение 20](#_Toc516320573)

[Список использованных источников 31](#_Toc516320574)

**ВВЕДЕНИЕ**

С развитием технологий и существенным увеличением мощности вычислительных систем, стали актуальными задачи автоматизации процессов и создания искусственного интеллекта (ИИ). Главной целью ИИ является решение задач, возникающих на практике.

Одним из активно развивающихся направлений ИИ являются искусственные нейронные сети (ИНС), работающие по принципу биологических нейронных сетей и представляющие из себя систему взаимодействующих между собой искусственных нейронов. Среди основных областей применения ИНС можно выделить: распознавание объектов, обработка звуковой и текстовой информации, прогнозирование, принятие решений, оптимизация, анализ данных.

В данной курсовой работе будет рассмотрена программная реализация сверточной нейронной сети, задачей которой является распознавание определенных объектов на первом изображении и обнаружение изменения положения этих объектов на втором изображении.

Целью курсовой работы является изучение концепции ИНС, разбор видов ИНС, способы обучения ИНС, ознакомление с библиотекой KERAS для работы с ИНС [3].

**1 Введение в искусственные нейронные сети**

**1.1 История развития ИНС**

Считается, что теория нейронных сетей как научное направление впервые была обозначена в классической работе Уорренна Мак-Коллока и Уолтера Питтса в 1943 г., в которой утверждалось, что практически любую логическую операцию можно реализовать с помощью простой нейронной сети. Продолжая исследования в этой области, в 1958 г. Фрэнк Розенблатт изобретает однослойный перцептрон являющийся одной из первых моделей нейронной сети и демонстрирует его способность решать задачи классификации. Однако, как оказалось позже, перцептрон имел серьезные ограничения. В 1969 году Марвин Ли Минский публикует формальное доказательство о неспособности перцептрона решать некоторые задачи, написав об этом целую книгу. Лишь спустя 20 лет, нейронные сети стали активно использоваться, что связано с энергетическим подходом Джона Хопфилда и созданием алгоритма обратного распространения ошибки впервые предложенного Вербосом и независимо разработанного рядом других авторов. Алгоритм получает известность благодаря Румельхарту в 1986 г., который активно используется в наше время.

С середины 80-х годов теория нейронных сетей получила еще больший импульс исследования в связи с появлением высокопроизводительных персональных компьютеров [1].

**1.2 Проблемы, решаемые в рамках ИНС**

На сегодняшний момент, круг задач, которые могут быть решены с использованием нейронных сетей достаточно велик. К самым распространенным можно отнести:

Распознавание образов и классификация**.** Суть заключается в распознавании входного образа (символов текста, изображения, речевых сигналов и т.д.) и указании его принадлежности к определенному классу. Во время обучения нейронной сети, на вход ей подают вектор значений признаков образа, с указанием его класса. После обучения, ей можно предъявлять неизвестные ранее образы и получать ответ о принадлежности к определенному классу. В такого рода задачах, устанавливается соответствие между выходным слоем ИНС и классом, который он представляет.

Задачи кластеризации**.** Задачи кластеризации, известные так же как классификация образов “без учителя”, не имеют обучающей выборки с метками классов. Их принцип работы основан на выявлении закономерностей и подобия между образами и размещении этих образов в один кластер или категорию [3].

Прогнозирование**.** Способность нейронных сетей к предсказанию результатов обусловлена обобщением и выделением общих зависимостей между входными и выходными данными. После обучения, сеть способна предсказать будущее значение некоторой последовательности, основываясь на ее предыдущих изменениях. Для того что бы ИНС могла дать максимально точный ответ, необходимо обучать ее на данных, в которых действительно прослеживается какая-то зависимость. В противном случае, прогнозирование вероятнее всего не даст никаких результатов.

Аппроксимация функций**.** Задача аппроксимации ИНС, заключается в нахождении оценки неизвестной функции, по ее значениям. Точность аппроксимации зависит от выбора структуры нейронной сети [3].

Оптимизация**.** Способы оптимизации с использованием нейронных сетей, подразумевают нахождение такого решения, которое удовлетворяет системе условий и минимизирует целевую функцию. В качестве примера можно рассмотреть алгоритм, реализующий некую последовательность действий, который можно полностью заменить функционированием нейронной сети, используя правила, по которым он составлен [3].

**1.3 Структура искусственного нейрона**

Искусственный нейрон является элементарной структурной единицей искусственной нейронной сети и представляющей из себя упрощенную моделью биологического нейрона. На рисунке 1 изображен нейрон, входами которого могут быть либо входные данные, либо выходы от такого же нейрона. Входы соединены с ячейкой нейрона S с помощью синаптических связей. Каждый синапс имеет свой вес, при передаче в ячейку нейрона входного параметра, он соответственно умножается на вес, т.е. xi\*wi. Состояние нейрона S, определяется в виде формулы:

 (1)

Выход нейрона, есть функция его состояния y=f(S).

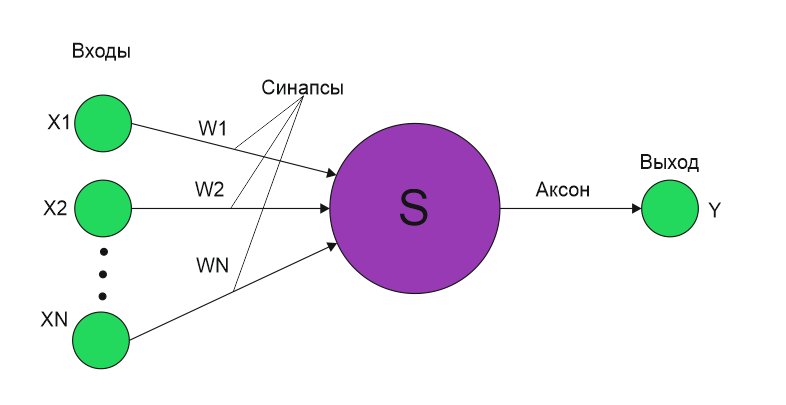


Рисунок 1 – Визуализация искусственного нейрона

Функция f называется функцией активации. Такие функции не могут быть линейными, поскольку нейронные сети с линейной функцией активации эффективны только на одном уровне, независимо от того, насколько сложна их структура. Одной из наиболее распространенных функций является нелинейная функция с насыщением, называемая логистическая функция или сигмоида:

 (2)

Параметр a отвечает за пологость функции. Чем он меньше, тем более пологим становится сигмоида. Следует отметить, что данная функция дифференцируема на всей оси абсцисс, что является необходимым

условием в некоторых алгоритмах обучения нейронных сетей. Кроме того, она обладает свойством усиливать слабые сигналы лучше, чем большие, а также предотвращает насыщение от больших сигналов так как они соответствуют областям аргументов, где сигмоида имеет пологий наклон (Рисунок 2) [2].

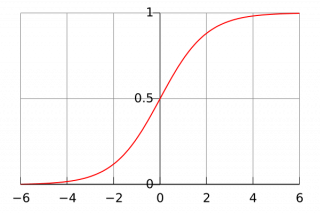


Рисунок 2 – Сигмоида

**1.4 Архитектура нейронной сети**

Можно выделить несколько основных типов нейронных сетей: Многослойные сети. В многослойных сетях один или несколько нейронов объединяются в слои (Рисунок 3). Слой – совокупность нейронов, на вход которых подается один и тот же общий сигнал. В сетях такого типа, внешние входные данные подаются на входы нейронов первого слоя, а выходные данные являются результатом последнего выходного слоя. Кроме входного и выходного слоев, в многослойных сетях так же присутствует один или несколько скрытых слоев. Связи выходов нейронов от некоторого слоя i к некоторому слою i+1 называют последовательным.

Полносвязные нейронные сети. В полносвязной нейронной сети каждый нейрон передает свой сигнал остальным нейронам (Рисунок 4).

Выходными сигналами, могут быть все или некоторые сигналы нейронов после нескольких тактов функционирования. Все входные сигналы, подаются на вход всем нейронам [6].

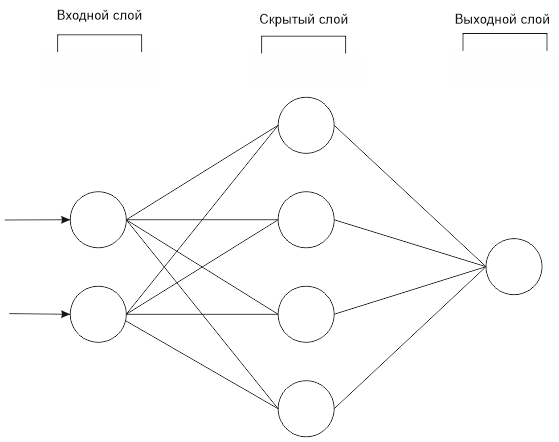


Рисунок 3 – Многослойная нейронная сеть

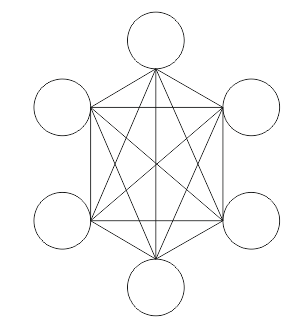


Рисунок 4 – Полносвязная нейронная сеть

Каждая нейронная сеть имеет не только свою архитектуру, но и тип, который лучше подходит для решения конкретной задачи. К примеру, сверточные нейронные сети (convolutional neural network CNN) гораздо лучше справляется с распознаванием образов и проблемами компьютерного зрения. Ее отличие от других типов ИНС заключается в том, что каждый фрагмент изображения умножается на матрицу (ядро) свертки поэлементно, а результат суммируется и записывается в аналогичную позицию выходного изображения (Рисунок 5). Если каждый пиксель изображения рассматривался бы отдельно, это привело бы сеть к быстрому переобучению и ее способность распознавания образов была бы точна только на обучающей выборке [7].

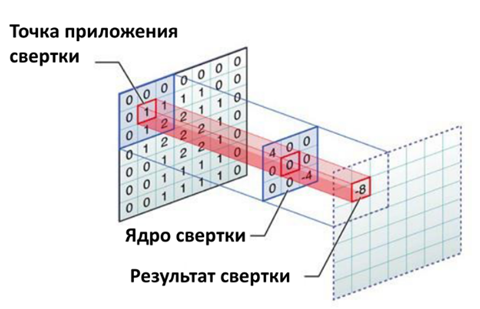


Рисунок 5 – Сверточная нейронная сеть

Существуют так же развертывающие нейронные сети (deconvolutional networks DN), так же называемые обратными графическими сетями и являющиеся обратными к CNN (Рисунок 6). Их задача генерировать изображения по заданным признакам. К примеру, при передаче сети слова “кот” она должна будет сгенерировать изображения похожие на котов.

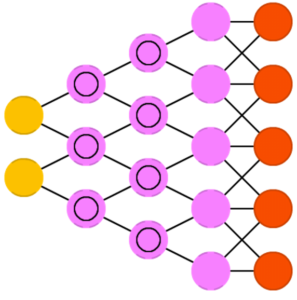


Рисунок 6 – Развертывающаяся нейронная сеть

Для предсказания слов в предложении, предсказании следующего числа в последовательности, выделения главной мысли текста, генерации новой информации похожей на данную используются рекуррентные нейронные сети (recurrent neural network RNN). В рекуррентных нейронных сетях нейроны обмениваются информацией между собой, к примеру, в добавок к новому кусочку входящих данных нейрон так же получает информацию о предыдущем состоянии сети (Рисунок 7). Таким образом, в сети реализуется так называемая “память”, что принципиально меняет характер ее работы и позволяет анализировать любые последовательности данных.

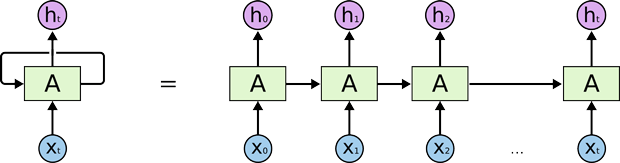


Рисунок 7 – Рекуррентная нейронная сеть

Самыми простыми нейронными сетями, являются сети прямого распространения (feedforward neural network FFN), которые взяты за основу для создания многих других сетей. Данные сети прямолинейны и передают информацию от входа к выходу. Нейроны каждого слоя не связаны между собой, а соседние слои обычно полностью связаны. Вид FFN можно рассматривать как многослойную сеть (Рисунок 3) [7].

**1.5 Обучение нейронных сетей**

Процесс обучения нейронной сети является необходимым, для ее способности выполнять поставленные задачи. Этот процесс может быть рассмотрен как настройка архитектуры сети и весов связей посредством моделирования среды, в которую эта сеть встроена. Свойство нейронной сети обучаться на примерах позволяет упростить задание условий для решения конкретной задачи, по сравнению с системами, которые следуют набору определенных правил, составленных экспертами. Разделяют алгоритмы обучение с учителем и без учителя [3].

Процесс обучения с учителем подразумевает предъявление сети выборки обучающих примеров. Каждый образец подается на вход нейронной сети, затем проходит процедуру обработки внутри ИНС. После вычисления выходного сигнала ИНС сравнивает полученный результат с соответствующим значением целевого вектора, представляющего собой требуемый выход сети. Вычислив ошибку, происходит изменение весовых коэффициентов связей внутри сети по выбранному алгоритму. Веса подстраиваются под каждый вектор до тех пор, пока ошибка по всему массиву входных данных не достигнет заданного уровня.

Обучение без учителя не требует знания правильного ответа на каждый пример обучающей выборки. В этом случае раскрывается внутренняя структура данных или корреляция в системе данных, что позволяет распределить образы по категориям.

В зависимости от решаемой задачи в обучающей выборке используются те или иные типы данных и различные размерности входных/выходных сигналов. Входные данные примеров обучающей выборки - изображения, таблицы чисел, распределения. Типы входных данных - бинарные (0 и 1), биполярные (-1 и 1) числа, целые или действительные числа из некоторого диапазона. Выходные сигналы сети - вектора целых или действительных чисел. Для решения практических задач часто требуются обучающие выборки большого объема. Из-за жестко ограниченного объема оперативной памяти компьютера разместить в ней большие обучающие выборки невозможно. Поэтому выборка делится на страницы - группы примеров. В каждый момент времени лишь одна страница примеров располагается в памяти компьютера, остальные - на жестком диске. Страницы последовательно загружаются в память компьютера. Обучение сети происходит по всей совокупности страниц примеров, по всей обучающей выборке.

В настоящее время отсутствует универсальная методика построения обучающих выборок. Набор обучающих примеров формируется по усмотрению пользователя программы моделирования нейронных сетей индивидуально для каждой конкретной решаемой задачи [4].

Если в необученную нейронную сеть ввести входной сигнал одного из примеров обучающей выборки, то выходной сигнал сети будет существенно отличаться от желаемого выходного сигнала, определенного в обучающей выборке. Функция ошибки численно

определяет сходство всех текущих выходных сигналов сети и соответствующих желаемых выходных сигналов обучающей выборки. Наиболее распространенной функцией ошибки является среднеквадратичное отклонение. Однако предложены и другие функции ошибки.

Для обучения нейронных сетей могут быть использованы различные алгоритмы. Можно выделить две большие группы алгоритмов - градиентные и стохастические. Градиентные алгоритмы обучения сетей основаны на вычислении частных производных функции ошибки по параметрам сети. Среди градиентных различают алгоритмы первого и второго порядков. В стохастических алгоритмах поиск минимума функции ошибки ведется случайным образом [2].

При обучении сетей, как правило, используется один из двух следующих критериев останова: останов при достижении некоторого малого значения функции ошибки, останов в случае успешного решения всех примеров обучающей выборки.

Перед обучением выполняется инициализация нейронной сети, то есть присваивание параметрам сети некоторых начальных значений. Как правило, эти начальные значения - некоторые малые случайные числа.

Для формирования обучающих выборок, инициализации и обучения в программах моделирования нейронных сетей используются специальные процедуры. Возможность использования многостраничного обучения является очень важной при решении практических задач с помощью нейронных сетей, моделируемых на обычных компьютерах.

Обучение - это итерационная процедура, которая при реализации на обычных компьютерах, требует значительного времени. Алгоритмы обучения существенно различаются по скорости сходимости. Одной из

самых важных характеристик программ для моделирования нейронных сетей является скорость сходимости алгоритма (или алгоритмов) обучения, которые реализованы в программе [2].

Теория обучения рассматривает три фундаментальных свойства, связанных с обучением по примерам: емкость, сложность образцов и вычислительная сложность. Под емкостью понимается, сколько образцов может запомнить сеть, и какие функции и границы принятия решений могут быть на ней сформированы. Сложность образцов определяет число обучающих примеров, необходимых для достижения способности сети к обобщению. Слишком малое число примеров может вызвать "переобученность" сети, когда она хорошо функционирует на примерах обучающей выборки, но плохо - на тестовых примерах, подчиненных тому же статистическому распределению. Известны 3 основных типа правил обучения: коррекция по ошибке, машина Больцмана и правило Хебба.

Правило коррекции по ошибке. При обучении с учителем для каждого входного примера задан желаемый выход d. Реальный выход сети y может не совпадать с желаемым. Принцип коррекции по ошибке при обучении состоит в использовании сигнала (d-y) для модификации весов, обеспечивающей постепенное уменьшение ошибки. Обучение имеет место только в случае, когда персептрон ошибается. Известны различные модификации этого алгоритма обучения [2].

Обучение Больцмана**.** Представляет собой стохастическое правило обучения, которое следует из информационных теоретических и термодинамических принципов. Целью обучения Больцмана является такая настройка весовых коэффициентов, при которой состояния видимых нейронов удовлетворяют желаемому распределению вероятностей. Обучение Больцмана может рассматриваться как специальный случай коррекции по ошибке, в котором под ошибкой понимается расхождение корреляций состояний в двух режимах.

Правило Хебба**.** Самым старым обучающим правилом является постулат обучения Хебба. Хебб опирался на следующие нейрофизиологические наблюдения: если нейроны с обеих сторон синапса активизируются одновременно и регулярно, то сила синаптической связи возрастает. Важной особенностью этого правила является то, что изменение синаптического веса зависит только от активности нейронов, которые связаны данным синапсом. Это существенно упрощает цепи обучения.

Для того что бы проверить навыки, приобретённые нейронной сетью в процессе обучения, используется имитация функционирования сети. В сеть вводится некоторый сигнал, который, как правило, не совпадает ни с одним из входных сигналов примеров обучающей выборки. Далее анализируется получившийся выходной сигнал сети. Тестирование обученной сети может проводиться на одиночных входных сигналах, либо на контрольной выборке, которая имеет структуру, аналогичную обучающей выборке [2].

**1.5.1 Алгоритм обратного распространения ошибки**

Алгоритм обратного распространения ошибки (back propagation) относится к методу обучения с коррекцией по ошибке и, как правило, применяется к многослойным перцептронам. Это метод обучения "с учителем", при котором "наставник" обучает сеть, также как ребенка обучают читать и писать. При обучении на входной слой многократно подаются образы сигналов, распознаванию которых нейронная сеть должна быть обучена, и корректируются веса нейронов для достижения

желаемого выходного сигнала. Для улучшения качества распознавания образы, подаваемые на входной слой, могут быть слегка изменены (добавлен шум и т.п.). Детально процедура обучения выглядит так:

1) Выборка входных данных (множество образов, классифицированных учителем) разбивается на две: обучающую и контролирующую последовательности. Обычно обучающая последовательность содержит больше образов, чем проверочная.

2) Производится инициализация всех весов, включая пороговые, небольшими случайными величинами (обычно в диапазоне [-1; +1]). Это определяет начальную точку на поверхности ошибок для метода градиентов, позиция может оказаться решающей для сходимости сети.

3) Производится прямой проход сети для первого образа из обучающей выборки от входного слоя через скрытые слои к выходному слою: каждый нейрон суммирует произведения входов на веса и выдает результат функции активации, примененной к этой сумме, на нейроны следующего уровня.

4) Вычисляется разность между действительным и желаемым выходным значением каждого нейрона выходного слоя. При их несовпадении имеет место ошибка в распознавании (классификации) образов.

5) Производится процедура обратного распространения этих ошибок по связям от выходных нейронов к входным и определяются ошибки для каждого нейрона. Рассмотрим эту процедуру подробнее. Предположим, выходное значение нейронной сети было 0.5, а желаемое 0. Пусть ошибка определяется по формуле:



Где E – ошибка сети, yi – желаемое значение на i-ом выходе нейронной сети, di – выходное значение сети на i-ом выходе.

В данном случае, на выходе ошибка будет равна E = 0.5-0 = 0.5. Если рассматривать в качестве функции активации сигмоид, то расчет Δw, на который необходимо сдвинуть веса синаптических связей предпоследнего последнего слоя, будет производиться по формуле:



Здесь sigmoid(x)dx – производная функции активации, равная sigmoid(x)(1-sigmoid(x)). Под x подразумевается значение, которое было получено путем суммирования всех выходных значений нейронов предпоследнего слоя, соответственно умноженных на wi. Завершающий шаг коррекции весов синаптических связей между последним и предпоследним слоем будет производиться по следующей формуле:



mi – выходные значения i нейрона предпоследнего слоя

l – параметр отвечающий за скорость обучения

После того как веса посчитаны, мы можем посчитать ошибку выходного значения mi по формуле E = wi\*Δw. Зная ошибку каждого нейрона на предыдущем слое, применяя формулы, которые использовались выше, можем найти отклонения весов и значения нейронов на предыдущих слоях.

6) Вновь производится прямой проход сети уже для очередного образа обучающей выборки...

Шаги 3-6 повторяются до достижения некоторого критерия,

например, достижения ошибкой установленного предела.

По завершению обучающей фазы сеть проверяется при помощи контролирующей последовательности, содержащей образы, не предъявленные ранее. (При этом не производится коррекция весов, а лишь вычисляется ошибка). В случае, если качество работы найдено удовлетворительным, сеть считается готовой к работе. В противном случае сеть подвергается повторному обучению, при котором возможно изменение некоторых параметров (начальные веса, количество нейронов в скрытых слоях, дополнительные обучающие образы и т.п.) [2].

**1.6 Некоторые проблемы ИНС**

Одной из главных проблем нейронных сетей является необходимость в очень больших объемах данных. Несколько лет назад, были продемонстрированы алгоритмы, способные распознавать образы на изображениях лучше, чем человек. Что бы превзойти человека, сетям пришлось изучить более 1,2 миллиона изображений, в то время как ребенок, может научиться определять новый объект или животное после одной увиденной фотографии. Практически для любой задачи распознавания образов, искусственным нейронным сетям необходимо в сотни тысяч раз больше информации чем человеку.

Другой проблемой, является не приспособленность нейронных сетей к мультизадачности. Современные алгоритмы предназначены только для решения одной задачи. ИНС можно научить распознавать собак или создавать музыку. Но на сегодняшний момент, не существует таких сетей, которые могли бы выполнять обе эти задачи. Если рассмотреть нейронную сеть с фиксированным количеством слоев и

нейронов в каждом слое, то при постоянном увеличении, к примеру, обучаемых образов, рано или поздно нейронная сеть перестанет поддаваться обработке. Или если уже обученную нейронную сеть распознавать лица, начать обучать распознавать кошек, то она начнет забывать про лица для освобождения памяти для новой информации [5].

В 2016 году был проведен эксперимент. Исследователи хотели определить, на какую часть изображения “смотрит” нейронная сеть что бы выполнить задачу. Они показали нейронной сети фотографию спальни и спросили у нее “Что висит на окнах?”. Вместо того, чтобы посмотреть на окна, ИНС стала смотреть на пол, после чего на кровать, из чего сделала вывод, что на окне висят занавески. Дело в том, что невозможно посмотреть “вглубь” нейронных сетей, для того что бы понять, как они работают.

Так же, одной из проблем нейронных сетей является ее правильное проектирование, к примеру, неизвестно сколько слоев необходимы для данной задачи, сколько нужно выбрать элементов в каждом слое, как сеть будет реагировать на данные, которые не включены в обучающую выборку (какова способность сети к обобщению) и какой размер обучающей выборки необходим для способности сети к обобщению. Чаще всего, данные параметры определяются путем проб и ошибок. Существующие теоретические результаты дают лишь слабое представление о том, какими должны быть эти параметры [5].

**2 Создание нейронных сетей**

**2.1 Среда разработки искусственных нейронных сетей**

В последние несколько лет наблюдается взрыв интереса к нейронным сетям. Все больше и больше материала про нейронные сети на разных языках можно найти на просторах интернета. Для упрощения создания ИНС стали создаваться различные фреймворки, в которых уже реализована аппаратная часть, отвечающая за вычисление функций ошибок, представление нейронов и их синаптических связей между собой, создание архитектур нейронных сетей, их типов и много другого. Более того, стали появляться программные средства, позволяющие создавать сложные нейронные сети без необходимости писать код.

Одним из лидирующих языков в создании искусственных нейронных сетей является скриптовый язык python. Из-за своей простоты в использовании и большим количеством библиотек для нейронных сетей написанных для него он представляет наибольший интерес в создании ИНС.

Самыми часто используемыми библиотеками являются TensorFlow, разработанная компанией Google, Theano основным разработчиком которой является группа машинного обучения в Монреальском университете и библиотека Keras, представляющую из себя надстройку над одним из вышеперечисленных фреймворков [1].

**2.2 Пример элементарной сети на Keras**

В качестве примера, рассмотрим программную реализацию нейронной сети на Keras (Рисунок 8)

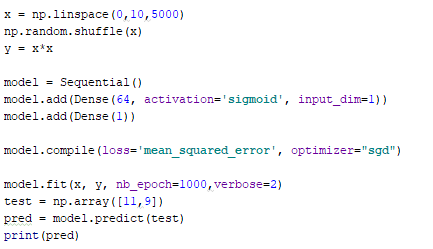
****

Рисунок 8 – Программная реализация сети на Keras

В качестве x в данном примере выступает numpy-массив, состоящий из 5000 элементов от 0 до 10. Данный массив перемешивается, после чего находится функция y=x\*x. Данные массивы можно рассматривать как входные и выходные данные для нейронной сети. С помощью библиотеки Keras, создаем модель Sequential, в которую поочередно добавляем слои из 64 нейронов с сигмоидной функцией активацией и еще один слой, состоящий из одного нейрона, который является выходным. Для обучения сети используется функция fit, которая принимает вектор входных параметров (x), вектор выходных параметров (y), количество эпох, в данном случае 1000. На каждом шаге (эпохе), ошибка уменьшается. После обучения сети, в качестве тестирования отправим ей два входных значения 11 и 9. Как мы можем заметить (Рисунок 9) полученные результаты 114 и 81 не являются точными. Все потому, что для задачи экстраполяции больше подходят рекуррентные нейронные сети, в то время как с задачей аппроксимации нейронная сеть справилась с небольшой погрешностью. Используя более сложные структуры нейронной сети, добавив больше эпох обучения и слоев с нейронами, можно добиться более точных результатов.

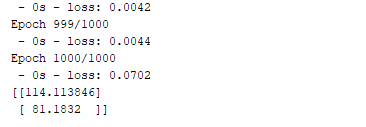


Рисунок 9 – обучение сети и получение тестовых результатов

Данный пример, демонстрирует лишь малую часть всех возможностей библиотеки Keras. Его инструментарий позволяет так же создавать рекуррентные и сверточные нейронные сети [8].

**3 Программная реализация нейронной сети**

**3.1 Постановка задачи**

Даны два изображения. Необходимо используя нейронную сеть распознать на первом изображении руки и лица людей, распознать руки и лица людей на втором изображении, и определить, было ли изменение позиций распознаваемых объектов на первом изображении с позициями на втором изображении соответственно на заданную величину. Если позиция изменилась, оповестить об этом в программе.

Данную задачу можно рассматривать как детектор движения заданных объектов и в дальнейшем улучшить до захвата изображения с камер и сравнением двух следующих друг за другом кадров.

**3.2 Выбор среды разработки**

Для создания данного проекта планировалось использование Tensorflow object detection API. Для его функционирования, необходимо выбрать из уже предоставленных предобученных сетей выбрать наиболее подходящую по скорости и точности распознавания. Так же, предварительно необходимо создать датасет, содержащий метки входных изображений с позиционированием каждого распознаваемого объекта. Для создания датасета использовалась программа labelimg, с помощью которой были созданы xml файлы с разметкой рук и лиц людей на изображении. После чего, данные файлы были преобразованы с помощью python скрипта в файлы записи, используемые Tensorflow object detection API. На выбор, данная программа предоставляет большой спектр нейронных сетей, которые отличаются своей

архитектурой. В качестве примера, была использована сверточная сеть, с названием “[ssd\_mobilenet\_v2\_coco](http://download.tensorflow.org/models/object_detection/ssd_mobilenet_v2_coco_2018_03_29.tar.gz)”. Она является самой быстрой из всех имеющихся по скорости, но самой неточной по распознаванию объектов. В большинстве случаев, ее используют для распознавания объектов в реальном времени на камерах мобильных телефонов. К данной сети, прилагается конфигурационный файл для ее настройки, где можно указать такие параметры как скорость обучения, функции активации нейронов, способ обучения нейронной сети. После настройки, через командную строку был запущен процесс обучения на данных, которые были подготовлены ранее с помощью labelimg. Несмотря на то, что все параметры для обучения были установлены для минимальной затраты ресурсов компьютера, процесс обучения нейронной сети по подсчетам мог бы занять продолжительное время. На каждый шаг обучения нейронной сети в среднем уходило по 60 секунд. Для нормального функционирования нейронной сети необходимо было примерно 20000 шагов, что заняло бы приблизительно 330 часов, поэтому, данный способ реализации не подходил для выполнения задачи.

Для создания программы будет использован фреймворк Keras, для Python, позволяющий создать нейронную сеть и избежать проблем с большим периодом ее обучения.

**3.3 Описание алгоритма реализации**

Любое изображение, может быть представлено в виде двухмерного массива, индексами которого являются координаты пикселей изображения, а их значениями массив, состоящий из 3 элементов (R, G, B), в диапазоне 0-255. Таким образом, каждой точке изображения однозначно соответствует три числа, определяющие ее цвет.

В дальнейшем, с помощью библиотеки Keras планируется создать модель сверточной нейронной сети Sequential, которая будет обучена на обучающей выборки созданной в программе labelimg, которая упрощает работу выделения объектов на изображении. После обучения нейронной сети, с помощью библиотеки OpenCV (библиотека компьютерного зрения) будет производиться преобразование и загрузка данных изображений в оперативную память компьютера. Изображения будут передаваться на вход обученной нейронной сети, которая будет выделять заданные объекты. Данные объекты будут храниться в массиве, каждый элемент которого будет иметь позицию x и y, высоту и ширину h и w соответственно. Помещая в массив таким же образом данные второго изображения, все позиции объектов будут сравниваться с позициями первого изображения, а также будет учитываться их класс. Если, поэлементно сравнивая позиции объектов на первом и втором изображении будет выполняться условие:



Где x2, y2 – координаты объекта на первом изображении, x1, y1 – координаты предположительно того же объекта на втором изображении, E – максимальная величина отклонения объекта, то это будет означать, что объект был смещен.

**ЗАКЛЮЧЕНИЕ**

В курсовой работе были рассмотрены теоретические основы нейронных сетей. В ходе многократных попыток обучения ИНС, для реализации данной задачи было решено использовать библиотеку Keras для python, позволяющей максимально быстро и просто создавать нейронные сети разных типов и архитектур.

Нейронные сети занимают лидирующие места в области искусственного интеллекта. Благодаря им, многие задачи могут быть решены без конструирования алгоритмов. Но для того, чтобы нейронные сети могли функционировать хотя бы приблизительно как биологические, им предстоит пройти очень долгий процесс эволюции. Исключительно из интереса к философии и мыслительной деятельности человека, я выбрал данную тему, в которую в будущем постараюсь внести свой вклад.

**СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

1 История возникновения нейронных сетей [Электронный ресурс] - https://neuronus.com/history/5-istoriya-nejronnykh-setej.html(25.12.2018)

2 [Нейронные сети: полный курс. С. Хайкин. 2-е издание, Издательский дом Вильямс, 2008 - 1104 с.](https://www.google.com/search?client=avast&q=%D0%9D%D0%B5%D0%B9%D1%80%D0%BE%D0%BD%D0%BD%D1%8B%D0%B5+%D1%81%D0%B5%D1%82%D0%B8:+%D0%BF%D0%BE%D0%BB%D0%BD%D1%8B%D0%B9+%D0%BA%D1%83%D1%80%D1%81,+2-%D0%B5+%D0%B8%D0%B7%D0%B4%D0%B0%D0%BD%D0%B8%D0%B5&stick=H4sIAAAAAAAAAONgecQYyi3w8sc9YSmfSWtOXmN04-IKzsgvd80rySypFNLhYoOylLgEpHj00_UNjZLyLHOTklM0GKT4uFBElLia9q04xMbCwSjAwAMAd0aOTVoAAAA&sa=X&ved=2ahUKEwjhgp7Ay_bfAhXHp4sKHbwGCicQ6RMwC3oECAYQBA)

3 Neural Networks: A Systematic Introduction. [R. Rojas](https://www.amazon.com/s/ref=dp_byline_sr_book_1?ie=UTF8&text=Raul+Rojas&search-alias=books&field-author=Raul+Rojas&sort=relevancerank). Springer Science & Business Media - 502 с.

4 Нейронные сети для обработки информации. С. Осовский. Пер. с польского И.Д. Рудинского. — М.: Финансы и статистика, 2002. — 344 с

5 Проблемы искусственного интеллекта [Электронный ресурс] - https://ain.ua/2017/04/21/problemy-iskusstvennogo-intellekta(23.12.2018)

6 Лекции по нейронным сетям [Электронный ресурс] -https://www.intuit.ru/studies/courses/6/6/lecture/178(15.12.2018)

7 Разновидности нейронных сетей [Электронный ресурс] - https://tproger.ru/translations/neural-network-zoo-1(15.12.2018)

8 Создание нейронной сети на Python для прогнозирования [Электронный ресурс] - https://habr.com/post/327022(27.11.2018)