Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Новосибирский государственный технический университет»

Кафедра теоретической и прикладной информатики

**Отчет ПО ПРАКТИКЕ**

Учебная практика: ознакомительная практика

(наименование практики в соответствии с учебным планом)

Направление подготовки: 01.03.02 Прикладная математика и информатика

|  |  |
| --- | --- |
| Выполнил:  Студент Иванов В.В.  (Ф.И.О.)  Группа ПМ-92  Факультет ПМИ.  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  подпись  «2» июня 2022 г. | Проверил:  Руководитель от НГТУ Попов А.А.  (Ф.И.О.)  Балл: \_\_\_\_\_\_\_\_\_\_, ECTS\_\_\_\_\_\_\_\_\_\_\_\_\_,  Оценка \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  «отлично», «хорошо», «удовлетворительно», «неуд.»  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  подпись  «8» июня 2022 г. |

Новосибирск 2022

**СОДЕРЖАНИЕ**

[**ТЕОРЕТИЧЕСКОЕ ВВЕДЕНИЕ** 3](#_Toc105583811)

[Постановка задачи 3](#_Toc105583812)

[Функция активации 4](#_Toc105583813)

[Обучение нейронной сети 5](#_Toc105583814)

[Сверточная нейронная сеть 8](#_Toc105583815)

[**ОПИСАНИЕ НАБОРА ДАННЫХ** 11](#_Toc105583816)

[**ОПИСАНИЕ РАЗРАБОТАННОЙ МОДЕЛИ** 13](#_Toc105583817)

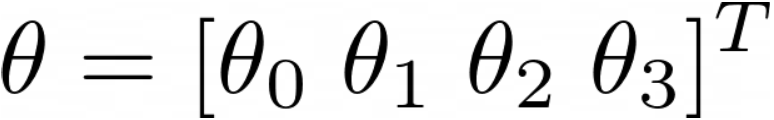
[**ЗАКЛЮЧЕНИЕ** 14](#_Toc105583818)

# **ТЕОРЕТИЧЕСКОЕ ВВЕДЕНИЕ**

# Постановка задачи

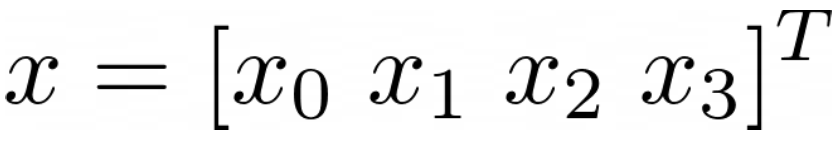
Нейронная сеть – это модель машинного обучения, представляющая из себя набор связанных между собой нейронов. Под нейроном здесь понимается максимально упрощенная модель нейрона головного мозга – функция, принимающая множество вещественных чисел и выдающая одно число в качестве результата.

Связь между нейронами определяется весом связи. Это можно понимать как её силу: число, проходящее через связь, умножается на вес этой связи. Веса являются параметром модели и заранее неизвестны исследователю.



Для подбора оптимальных весов модель необходимо обучить.

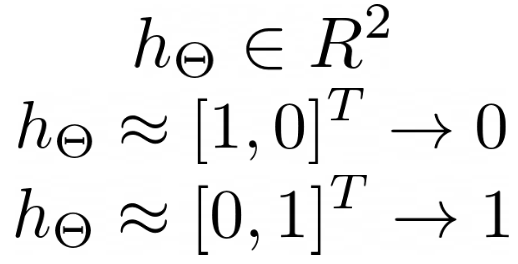
Нейронная сеть состоит из входного, внутренних (скрытых) и выходного слоев. Входной слой принимает вектор признаков:



Внутренние (скрытые) слои представляют собой “чёрный ящик” – сложную для понимания человеком стохастическую систему, в которой происходит запоминание признаков.

Выходной слой, основываясь на результатах работы нейронов внутренних слоев, выдает конечный результат, характер которого может варьироваться в зависимости от задачи. В общем случае это некоторая статистическая гипотеза о признаках *x* на параметрах *θ*.

Мы рассмотрим нейронную сеть в задаче бинарной классификации. В этом случае выходной слой будет состоять из двух нейронов. Активация первого нейрона будет означать отнесение моделью объекта к классу 0, второго – к классу 1:



Упрощенная модель такой нейронной сети изображена на рис. 1.

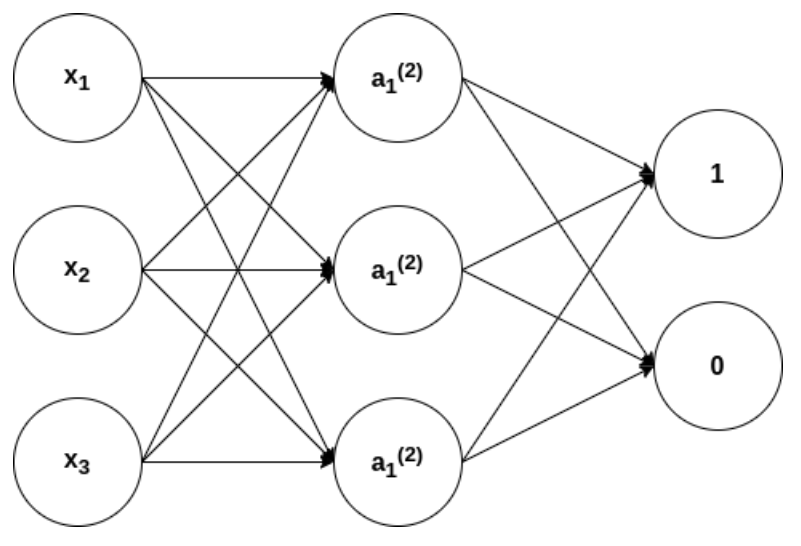


Рис. 1

В скобках здесь и далее указан номер слоя.

# Функция активации

Функция, заданная в каждом нейроне, называется функцией активации или передаточной функцией. В качестве параметра она принимает сумму всех чисел, переданных от других нейронов с учетом весов.

Наиболее популярной функцией активации является логистическая функция (сигмоида):

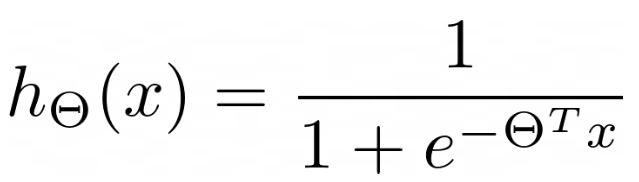


график которой имеет следующий вид (рис. 2):

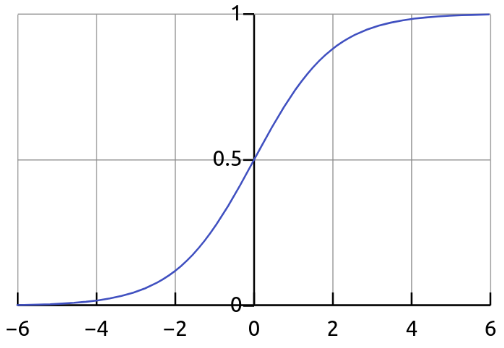


Рис. 2

Результат умножения входного значения нейрона на вес можно изобразить на графике этой функции. Если значение больше 0.5 – нейрон будет активирован.

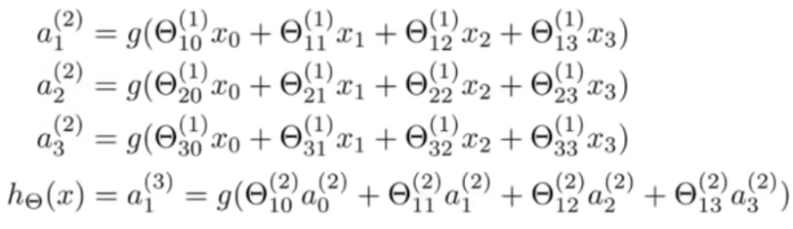
Различные передаточные функции используются для разных задач. Но существуют некоторые желательные свойства для всех функций. В частности, непрерывная дифференцируемость функции необходима для методов оптимизации на основе градиентного спуска, а монотонность – для гарантии выпуклости поверхности ошибок.

# Обучение нейронной сети

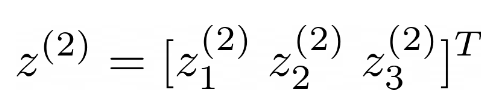
Обучение нейронной сети заключается в подборе такого вектора весов, который позволил бы модели запомнить все необходимые закономерности в данных (например, формы объектов на изображениях). При чтении очередного изображения нейронная сеть выполняет обход всех нейронов слева направо, задействуя те, сила связи которых является наибольшей.

По указанию исследователя этот процесс для одного объекта может быть произведен в несколько итераций. Как уже было сказано, результатом работы модели будет активация одного нейрона на последнем (выходном) слое – его значение будет являться результатом классификации.

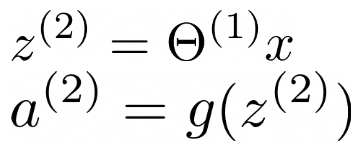
Опишем данный процесс более формально. Для этого выпишем выражения для каждого нейрона второго слоя (здесь и далее g – функция активации):



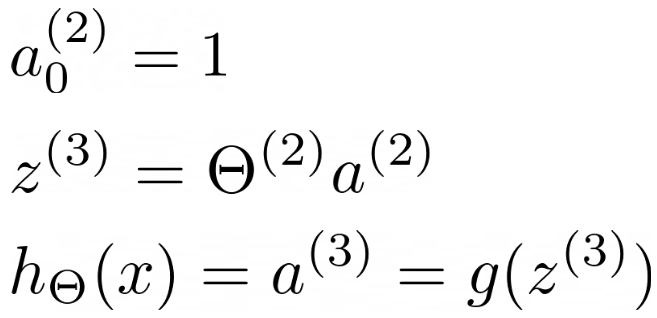
Перейдем к векторной записи. Для этого обозначим аргументы функции активации следующим образом:



Заметим, что новая переменная является суммой признаков с весами:



Добавляем смещение и представляем следующий слой в векторной форме:



Получили формулу логистической регрессии. Она считается для каждого слоя, однако в качестве обыкновенных признаков выступают значения из предыдущий слоев.

Этот процесс последовательного учёта весов на каждом слое называют прямым распространением (forward propagation). Итераций получаем на единицу меньше количества слоёв нейронной сети, так как в первом слое ничего считать не нужно.

Непосредственно обучение нейронной сети происходит во время итерации обратного обхода слоев (справа налево). Для этого используется метод обратного распространения ошибки (backpropagation) – измененной версии метода градиентного спуска для обновления весовых коэффициентов с помощью распространении сигналов ошибки от выходов сети к её входам. Напомним, что примерами «входов» и правильных «выходов» являются объекты из тренировочного набора данных.

Первоначально в нейронной сети веса расставлены случайным образом. После первой итерации (первого обхода вперед к выходному слою и обратно к источнику) нейронная сеть корректирует веса, исправляя реакцию нейронов на сигналы, поступающие по определенным связям. Затем производится необходимое количество итераций до тех пор, пока не будет получен результат.

За корректировку весов отвечает функция потерь (штрафная функция). Для нейронной сети эта функция практически полностью совпадает с функцей кросс-энтропии (Log Loss). Введем сначала её:

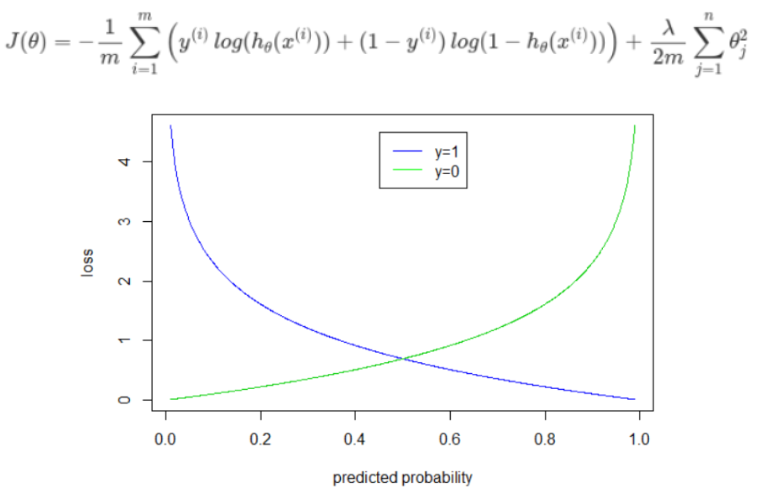
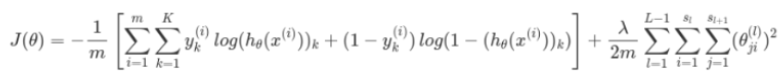


Рис. 3

Эта функция обладает интересными для нас свойствами, которые можно заметить, если построить её график (рис. 3). Во-первых, функцию логарифма довольно легко оптимизировать, поскольку она непрерывно убывает (возрастает). Во-вторых, функция кросс-энтропии сильно штрафует за уверенность классификатора в неверных ответах. То есть, ложноположительные ответы будут иметь очень высокую ошибку, что приводит к более высокой точности модели.

Модификация данной функции для обучения нейронной сети выглядит следующим образом:



где L - количество слоев нейронной сети, – количество нейронов на слое .

Минимизация функции потерь, как правило, осуществляется при помощи алгоритма градиентного спуска или его модификаций.

Таким образом, обучаясь, нейронная сеть на каждом новом слое создаёт собственные признаки, а веса связей позволяют ей запомнить информацию, повторив (на новых данных) путь по нужным нейронам к выходному слою.

Вышеописанная архитектура называется перцептроном – нейронной сетью прямого распространения, в которой все связи направлены строго от входных нейронов к выходным.

# Сверточная нейронная сеть

Описанная архитектура нейронной сети (перцептрон) может применяться для классификации изображений. Однако это крайне неэффективно. Главной проблемой данного подхода является потенциальная сложность нейронной сети: модели требуется отдельно обработать каждый пиксель изображения и каким-то образом выявить признаки.

Поэтому практически во всех случаях для решения задачи классификации изображений используется архитектура сверточной нейронной сети. Ключевым принципом данной архитектуры является процесс свертки – обход изображения окном определенного размера, позволяющий выделить признаки уже на первом слое нейронной сети. Это значительно ускоряет процесс обучения и точность модели. Процесс свертки показан на рис. 4.

Свертка происходит в отдельном сверточном слое. В отличии от рассмотренной ранее архитектуры, в сверточной нейронной сети при свертке используется лишь ограниченная матрица весов, которую мы называли окном свертки. Веса по-прежнему неизвестны и определяются в ходе обучения.

Окно двигается по всему обрабатываемому слою (в самом начале – непосредственно по входному изображению), формируя после каждого сдвига сигнал активации для нейрона следующего слоя с аналогичной позицией. То есть для различных нейронов выходного слоя используются одна и та же матрица весов, которую также называют ядром свёртки. Её интерпретируют как графическое кодирование какого-либо признака, например, наличие наклонной линии под определённым углом. Тогда следующий слой, получившийся в результате операции свёртки такой матрицей весов, показывает наличие данного признака в обрабатываемом слое и её координаты, формируя так называемую карту признаков (feature map).

Естественно, в сверточной нейронной сети набор весов не один, а целая гамма, кодирующая элементы изображения (например, линии и дуги под разными углами). При этом такие ядра свёртки не закладываются исследователем заранее, а формируются самостоятельно путём обучения сети методом обратного распространения ошибки. Обход каждым набором весов формирует свой собственный экземпляр карты признаков, делая нейронную сеть многоканальной (много независимых карт признаков на одном слое). Следует отметить, что при переборе слоя матрицей весов её передвигают обычно не на полный шаг (размер этой матрицы), а на небольшое расстояние. Так, например, при размерности матрицы весов 5×5 её сдвигают на один или два нейрона (пикселя) вместо пяти, чтобы не перешагнуть признак.

Второй компонент сверточных нейронных сетей – субдискретизирующий слой (слой пулинга). Как правило, этот слой располагается за сверточным. Его задача – уменьшение размерности сформированных карт признаков. Считается, что информация о факте наличия искомого признака важнее точного знания его координат, поэтому из нескольких соседних нейронов карты признаков выбирается максимальный и принимается за один нейрон уплотнённой карты признаков меньшей размерности. То есть, если на предыдущей операции свёртки уже были выявлены некоторые признаки, то для дальнейшей обработки настолько подробное изображение уже не нужно, и оно уплотняется до менее подробного. За счёт данной операции, помимо ускорения дальнейших вычислений и избавления от лишних деталей (что избавляет от эффекта переобучения), нейронная сеть становится более инвариантной к масштабу входного изображения.

Таким образом, сверточная нейронная сеть, как правило, состоит из нескольких чередующихся сверточных и субдискретизирующих слоев. Это позволяет осуществлять переход от конкретных особенностей изображения к более абстрактным деталям – вплоть до выделения понятий высокого уровня. Нейронная сеть сама вырабатывает необходимую иерархию абстрактных признаков, фильтруя маловажные детали и выделяя существенное. Эти данные объединяются и передаются на обычную полносвязную нейронную сеть, которая тоже может состоять из нескольких слоёв. При этом полносвязные слои уже утрачивают пространственную структуру пикселей и обладают сравнительно небольшой размерностью.

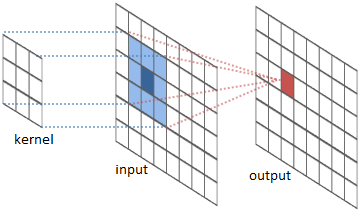


Рис. 4

# **ОПИСАНИЕ НАБОРА ДАННЫХ**

Набор данных состоит из 675 изображений. Все изображения цветные и имеют размер 128 на 128 пикселей. Названия каждого файла (изображения) начинается с номера класса (например, *0-124.png* или *1-5.png*).

Разметка набора данных заключается в разделении его на массивы *X* и *y*. Массив *X* содержит информацию о каждом пикселе каждого изображения и имеет размерность (n, 128, 128, 3), где 3 - это количество цветовых каналов (RGB), n - количество изображений в НД. Массив y содержит номер класса (который поставлен в соответствие каждому изображению) и имеет размерность (n, 1).

Затем массивы *X* и *y* разделяются на тренировочные и тестовые данные в пропорции 20/80. В результате получаем массивы следующих размерностей:

* X\_train (0.8n, 128, 128, 3)
* y\_train (0.8n, 1)
* X\_test (0.2n, 128, 128, 3)
* y\_test (0.2n, 1)

Перед обучением нейронной сети к НД применяется аугментация изображений при помощи функции ImageDataGenerator() класса keras.preprocessing.image. Для каждого изображения создается 10 копий (батчей) с различными визуальными эффектами, которые будут применяться случайным образом. Затем все копии помещаются в тренировочный НД (*X\_train*, *y\_train*). Генератор изображений имеет следующие параметры:

* horizontal\_flip=True (отражение по горизонтали)
* width\_shift\_range=0.15 (смещение по горизонтали)
* height\_shift\_range=0.15 (смещение по вертикали)
* rotation\_range=30 (вращение в пределах 30 градусов)
* ­brightness\_range=[0.5, 1.5] (изменение яркости в указанных пределах)
* zoom\_range=[0.9, 1.2] (приближение/отдаление в указанных пределах)
* channel\_shift\_range=15 (смещение цветовой гаммы)

Таким образом, после аугментации НД массивы *X\_train* и *y\_train* имеют размерности (10n, 128, 128, 3) и (10n, 1) соответственно.

Завершающим шагом предобработки данных является масштабирование (нормализация) признаков. Для этого каждый элемент массивов *X\_train* и *X\_test* делится на число 255.

# **ОПИСАНИЕ РАЗРАБОТАННОЙ МОДЕЛИ**

Модель состоит из трёх сверточных слоев, каждый из которых имеет ядро свертки размером 3 на 3 пикселя. Между ними осуществляется процесс нахождения подвыборки по максимуму (max pooling), снижающий размер изображения вдвое. После свертки осуществляется процесс случайного исключения нейронов с определенной вероятностью (dropout), выполняющий функцию регуляризации нейронной сети. Результат передается в два стандартных плотных (dense) слоя, связанных всеми нейронами, а затем – в выходной слой, количество нейронов которого соответствует количеству классов (2).

На каждом слое в качестве функции активации используется ReLU. В качестве алгоритма оптимизации используется Adam (модификация стохастического градиентного спуска). Обучение происходит в 10 эпох, размер батча – стандартный (весь НД). Функция потерь – кросс-энтропия (log loss).

# **ЗАКЛЮЧЕНИЕ**

По окончании выполнения практики были изучены все основные принципы, входящие в основу построения и обучения нейронных сетей, а также принципы, входящие в архитектуру сверточной нейронной сети. Был собран, обработан и размечен набор данных, изучены инструменты аугментирования изображений и приобретен опыт построение моделей глубокого обучения с нуля.

Результаты тестирования разработанной архитектуры нейронной сети показал, что сверточная нейронная сеть хорошо справляется с задачей бинарной классификации изображений, особенно в сравнении с обыкновенной архитектурой перцептрона.

Оценка модели на тестовых данных проводилась с использованием двух метрик классификации и показала следующие результаты:

* Стандартная метрика (accuracy): 95% точности
* F1-мера: 80% точности

Большая точность на стандартной метрике объясняется несбалансированностью набора данных: тренировочные данные относятся к тестовым приблизительно в соотношении 60/40.