

часть 2

ОБУЧЕНИЕ НЕЙРОСЕТЕЙ

Искусственные нейронные
сети для школьников

учебное пособие
для внеурочных занятий

Матвеева С.С.

Аннотация

Учебное пособие «Искусственные нейронные сети для школьников» является отдельным курсом для изучения IT направления в области создания нейронных сетей и рассчитано на учащихся средней и старшей школы. Изучение курса позволяет получить знания о искусственном интеллекте и нейронных сетях, сфере их применения, освоить первичные навыки по программированию и обучению простейшей нейросети.

Учебное пособие «Искусственные нейронные сети для школьников» рассчитан на среднюю и старшую школу и делится на три части:

- **1 часть** «Знакомство с ИИ и нейросетями»;
- **2 часть** «Обучение нейронных сетей, профессия нейропрограммист»;
- **3 часть** «Обучение и тестирование нейросети, профессия тестировщик».

Каждое пособие может быть самостоятельным курсом и использоваться для конкретных возрастных групп. Знания и умения полученные при изучении курса, помогут определиться с профессией будущего.

Данное учебное пособие является 2 частью. **2 Часть «Обучение нейросетей, профессия нейропрограммист»** познакомит Вас с простейшим видом искусственных нейронных сетей – персептронами, их устройстве. Вы научитесь создавать, обучать простейшие нейросети с помощью обучающей и тестовой выборки. Напишите обучающую программу для нейросети «Распознавание букв» на языке программирования Python, протестируете и проанализируете полученный результат.

Для изучения **2 Части «Обучение нейросетей, профессия нейропрограммист»** рекомендуется ознакомиться с **1 Частью «Знакомство с ИИ и нейросетями»**

Введение

Уважаемые школьники!

Мы живем во время стремительного развития технологий, уже сейчас на службе у человека есть различные роботы, выполняющие сложную и трудоемкую работу. Роботами управляет человек с помощью программного обеспечения и операторов, отслеживающих четкое выполнение поставленных задач.

Современные технологии развиваются настолько быстро, что в скором будущем роботы сами без контроля человека, смогут решать задачи, обучаться, развиваться, «думать», приносить пользу в различных сферах жизни: образовании, медицине, промышленности, в творчестве и т.д. Для этого роботам будет нужен искусственный «мозг». Искусственный мозг имеет название **Искусственная Нейронная Сеть (ИНС)**-это одно из направлений в разработке **ИИ (Искусственный интеллект)**.

ИНС – является упрощенной моделью биологической нейронной сети, представляющей собой совокупность искусственных нейронов, взаимодействующих между собой. Созданием ИНС занимаются нейропрограммисты. От нейропрограммистов зависит какими роботы будущего будут, что они будут уметь делать, будут ли они безопасны для человека, будут ли они жить с нами бок о бок в повседневной жизни.

При изучении курса «Искусственные нейронные сети для школьников» Вы узнаете направления и перспективы развития ИИ, технологии развивающие его, сферы использования, результаты применения, познакомитесь с профессией нейропрограммист.

ВСЕМ ЖЕЛАЮ УСПЕХА В ИЗУЧЕНИИ РАБОТЫ НЕЙРОННЫХ СЕТЕЙ!

2 Часть

«Обучение нейросетей, профессия нейропрограммист»

§ 1.

История персептрона

Словарь

- Первая статья о ИНС
- Персептрон

В далеком 1943 году Уоррен Мак-Каллок написал одну из первых статей на тему искусственной нейронной сети, в которой смоделировал работу мозга. А в 1960 году Фрэнк Розенблатт — известный американский учёный в области психологии, нейрофизиологии и искусственного интеллекта, представил миру первый “нейро-компьютер” “Марк-1”, он был способен распознавать некоторые буквы английского алфавита.

рис.1. Марк-1



Розенблатт предложил схему устройства, моделирующего процесс человеческого восприятия, и назвал его **«персептроном»** perceptio – восприятие.

Персептрон – это первая модель нейросети, а компьютер **«Марк-1»** первый нейрокомпьютер.

Марвин Ли Минский-американский учёный в области искусственного интеллекта, Написал книгу «Персептроны» (с Сеймуром Папертом), ставшую фунда-

ментальной работой для последующих разработок в области искусственных нейронных сетей. Привёл ряд своих доказательств теоремы сходимости перцептрона. Содержащаяся в книге критика исследований в этой области и демонстрация необходимых для этого вычислительных ресурсов считается причиной утраты интереса к искусственным нейронным сетям в академических статьях 1970-х годов.

«Наш математический анализ показал, почему увеличение размера перцептрона не приводит к улучшению способности решения сложных задач. Более того, в противоречие с общепринятым мнением, практически все теоремы могут быть применимы и к многослойным последовательным однонаправленным нейронным сетям. Хотя интересно, что никто этого так и не доказал, а Паперт и я перешли к следующим вопросам в этой области.»

§ 2.

Перцептрон

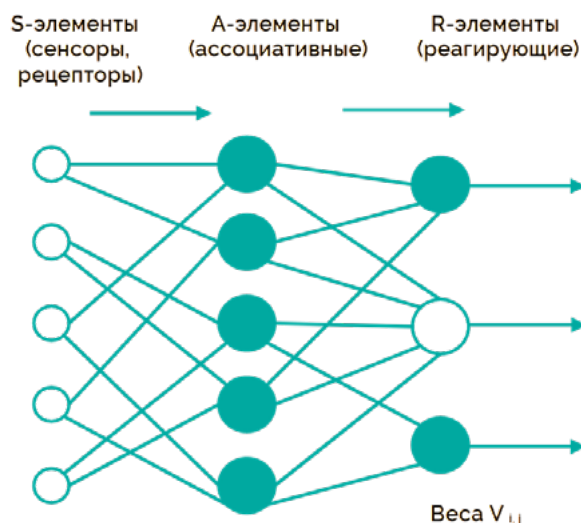
Словарь

- Перцептрон
- Принцип работы перцептрона

Перцептрон (Perceptron) — простейший вид нейронных сетей. В основе лежит математическая модель восприятия информации мозгом, состоящая из сенсоров, ассоциативных и реагирующих элементов.

Перцептрон состоит из трёх типов элементов, а именно сенсоров, ассоциативных элементов и реагирующих элементов поступающие от датчиков сигналы передаются ассоциативным элементам, а затем регулирующим элементам, таким образом перцептрон позволяет создать набор ассоциаций между входными стимулами и необходимой реакцией на выходе.

Рис. 2. Перцептрон



В биологическом плане это соответствует преобразованию, например, зрительной информации в физиологический ответ от двигательных нейронов. Согласно современной терминологии, перцептроны могут быть классифицированы как искусственные нейронные сети:

- с одним скрытым слоем;
- с пороговой передаточной функцией;
- с прямым распространением сигнала.

Принцип работы персептрона.

Первыми в работу включаются **S-элементы**. Они могут находиться либо в состоянии покоя (сигнал равен 0), либо в состоянии возбуждения (сигнал равен 1).

Далее **сигналы от S-элементов передаются A-элементам по так называемым S-A связям**. Эти связи могут иметь веса, равные только -1, 0 или 1.

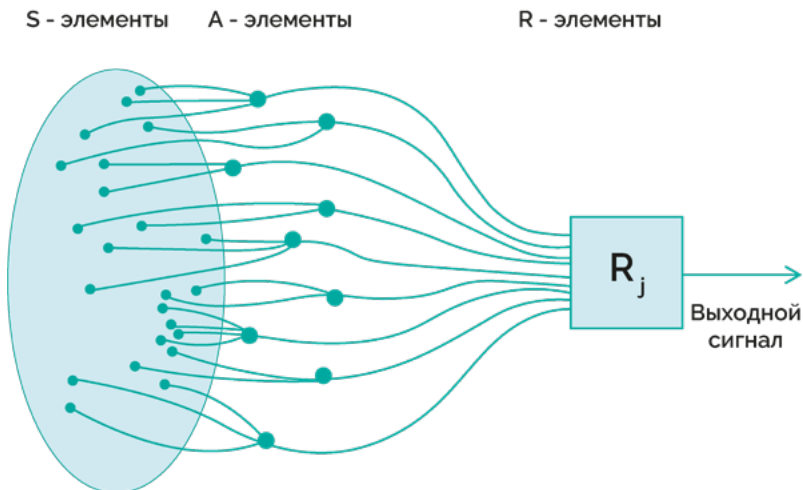
Затем сигналы от сенсорных элементов, прошедших по S-A связям попадают в **A-элементы**, которые еще называют ассоциативными элементами. Стоит заметить, что одному A-элементу может соответствовать несколько S-элементов.

Вывод: Если сигналы, поступившие на A-элемент, в совокупности превышают некоторый его порог θ , то этот A-элемент возбуждается и выдает сигнал, равный 1. В противном случае (сигнал от S-элементов не превысил порога A-элемента), генерируется нулевой сигнал.

A-элементы являются агрегаторами сигналов от сенсорных элементов. Например, у нас есть группа сенсоров, каждый из которых распознает кусок буквы «Д» на исследуемой картинке. Однако только их совокупность (то есть когда несколько сенсоров выдали сигнал, равный 1) может возбудить A-элемент целиком. На другие буквы A-элемент не реагирует, только на букву «Д». То есть он **ассоциируется с буквой «Д»**, поэтому **A-элементы назвали ассоциативными**.

Можно привести и другой пример. На самом деле ваши глаза состоят из невероятного количества **S-элементов (сенсоров)**, улавливающих падающий свет (около 140 000 000). И у вас какой-то A-элемент, который **распознает конкретную часть лица**. И вот вы увидели на улице человека. **Некоторые A-элементы, которые распознали конкретные части лица, возбуждаются.**

рис. 3. Персептрон сенсоры.



Далее сигналы, которые произвели возбужденные А-элементы, направляются к сумматору (R-элемент), действие которого вам уже известно. Однако, чтобы добраться до R-элемента, они проходят по А-R связям, у которых тоже есть веса. Однако здесь они уже могут принимать любые значения (в отличие от S-A связей).

Вывод: R-элемент складывает друг с другом взвешенные сигналы от А-элементов и, если превышен определенный порог, генерирует выходной сигнал, равный 1. Это означает, что в общем потоке информации от глаз мы распознали лицо человека.

Если порог не превышен, то выход персептрона равен -1. То есть мы не выделили лицо из общего потока информации.

Так как R-элемент определяет выход персептрона в целом, его назвали **реагирующим**.

Вопросы:

Как называется простейший вид нейронных сетей. В основе лежит математическая модель восприятия информации мозгом, состоящая из сенсоров, ассоциативных и реагирующих элементов?

1. Из каких типов элементов состоит математическая модель персептрона?
2. Что представляет собой R-элемент?
3. В каком случае S элементы возбуждаются?
4. В каком случае S элементы находятся в состоянии покоя?
5. Какие слова пропущены? R-элемент _____ друг с другом взвешенные сигналы от А-элементов и, если превышен определенный порог, генерирует выходной сигнал, равный _____. Это означает, что в общем потоке информации от глаз мы распознали лицо человека. Если порог не превышен, то выход персептрона равен _____. То есть мы не выделили лицо из общего потока информации.
6. Какие веса могут принимать S-A связи?
7. Почему А-элементы назвали ассоциативными?
8. Сигналы, проходящие по А-R связям имеют _____ и принимают любые значения.

§ 3.

Классификация перцептрона

Словарь

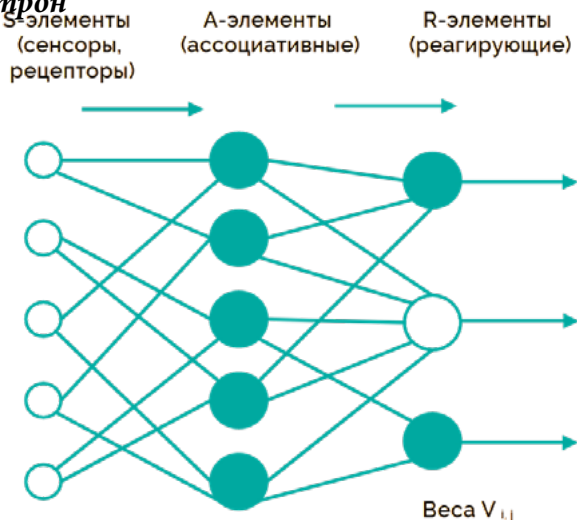
- Перцептрон с одним скрытым слоем
- Однослойный перцептрон
- Многослойный перцептрон

Типы перцептронов.

Перцептрон с одним скрытым слоем.

Перцептрон с одним скрытым слоем — перцептрон, у которого имеется только по одному слою S, A и R элементов. Слои A-элементов расположен между слоями S-элементов и R-элементов.

рис. 4. Перцептрон



Однослойный перцептрон.

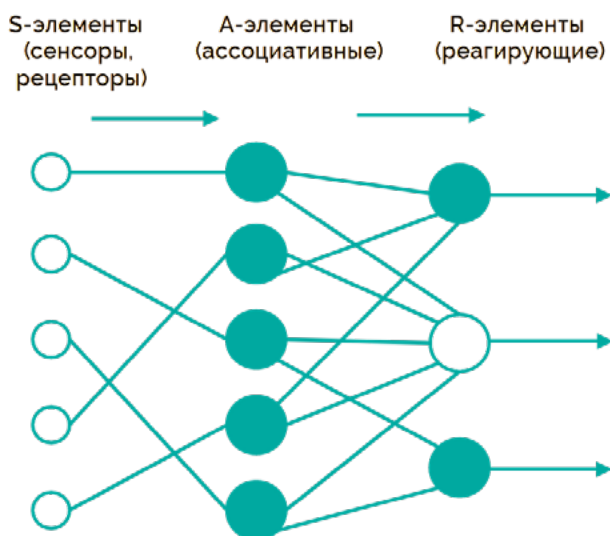
Однослойный перцептрон – ключевая особенность состоит в том, что каждый S-элемент однозначно соответствует одному A-элементу, все S-A связи имеют вес, равный +1, а порог A элементов равен 1.

Однослойный перцептрон — перцептрон, каждый S-элемент которого однозначно соответствует одному A-элементу, S-A связи всегда равны 1, а порог любого A-элемента равен 1.

Каждый S-элемент однозначно соответствует одному A-элементу, это означает,

что каждый сенсор может передавать сигнал только одному А-элементу. Так же любой другой S-элемент может передавать сигнал этому же А-элементу, т.е может быть ситуация когда несколько сенсоров передают сигнал на один А-элемент.

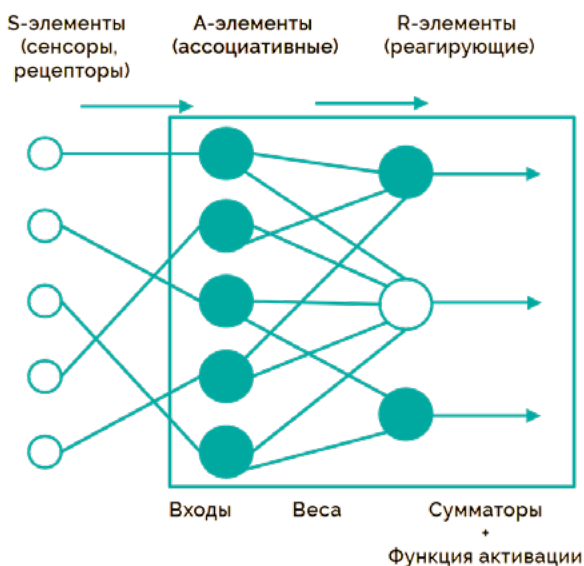
рис. 5. Однослойный перцептрон



S-A связи всегда имеют вес, равный единице, а порог А-элементов всегда равен +1.
С другой стороны нам известно, что сенсоры могут подавать сигнал равный только 0 или 1.

Рассмотрим первый S-элемент на

рис. 6. Однослойный перцептрон, функция активации



Допустим, он генерирует сигнал, равный единице. Сигнал проходит по S-A связи и не изменяется, так как любое число, умноженное на 1 равно самому себе. Порог любого A-элемента равен 1. Так как сенсор произвел сигнал, равный 1, то A-элемент однозначно возбуждился. Это означает, что он выдал сигнал, равный 1 (так как он тоже может генерировать только 1 или 0 на своем выходе). Далее этот единичный сигнал умножается на произвольный вес A-R связи и попадает в соответствующий R-элемент, который суммирует все поступившие на него взвешенные сигналы, и если они превышают его порог, выдает +1. В противном случае выход данного R-элемента равен -1.

Если не брать в расчет работу сенсорных элементов и S-A связей, то данная схема полностью соответствует схеме работы искусственного нейрона.

Вывод: Однослойный персептрон действительно представляет собой искусственный нейрон с небольшим отличием. В отличие от искусственного нейрона, у однослойного персептрона входные сигналы могут принимать фиксированные значения: 0 или 1. У искусственного нейрона на вход можно подавать любые значения. Однослойный персептрон может быть и элементарным персептроном, у которого только по одному слою S, A, R-элементов.

Многослойный персептрон.

Многослойный персептрон по Розенблатту — персептрон, у которого имеется более 1 слоя A-элементов.

Многослойный персептрон по Румельхарту — многослойный персептрон по Розенблатту, у которого обучению подлежат еще и S-A связи, а также само обучение производится по методу обратного распространения ошибки.

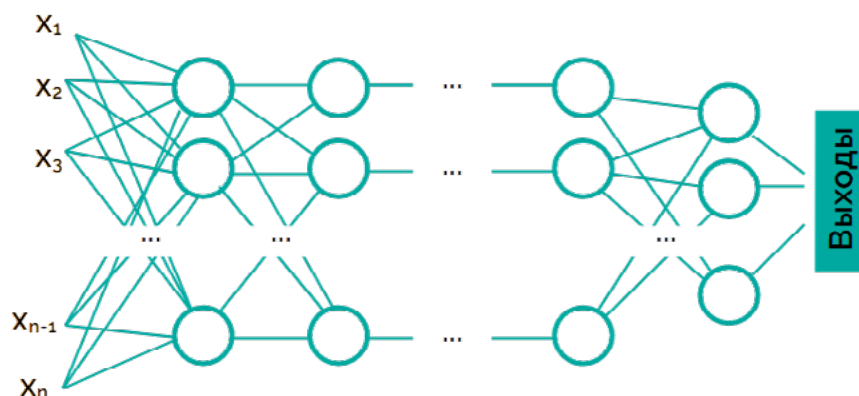
Под многослойным персептроном понимают два разных вида: многослойный персептрон по Розенблатту и многослойный персептрон по Румельхарту.

Многослойный персептрон по Розенблатту содержит более 1 слоя A-элементов.

Многослойный персептрон по Румельхарту является частным случаем многослойного персептрона по Розенблатту, с двумя особенностями:

- 1. S-A связи могут иметь произвольные веса и обучаться наравне с A-R связями.**
- 2. Обучение производится по специальному алгоритму, который называется обучением по методу обратного распространения ошибки.**

рис. 7. Многослойный перцептрон



Вывод: Многослойный перцептрон по Розенблатту содержит более 1 слоя А-элементов. по Румельхарту с двумя особенностями:

3. S-A связи могут иметь произвольные веса и обучаться наравне с A-R связями.
4. Обучение производится по специальному алгоритму, который называется обучением по методу обратного распространения ошибки.

Вопросы:

1. Что представляет из себя многослойный перцептрон?
2. С какими двумя особенностями многослойный перцептрон по Румельхарту является частным случаем?
3. Верно ли утверждение: Перцептрон с одним скрытым слоем — перцептрон, у которого имеется только по одному слою S, A и R элементов. слой A-элементов расположен между слоями S-элементов и R-элементов.?
4. Верно ли утверждение: Однослойный перцептрон действительно представляет собой искусственный нейрон с небольшим отличием. В отличие от искусственного нейрона, у однослойного перцептрона входные сигналы могут принимать фиксированные значения: 0 или 1. У искусственного нейрона на вход можно подавать любые значения. Однослойный перцептрон может быть и элементарным перцептроном, у которого только по одному слою S, A, R-элементов?
5. Сколько слоев содержит многослойный перцептрон по Розенблатту?

§ 4.

Практика распознавания букв

Словарь

- Персептрон с целыми значениями A-S связей

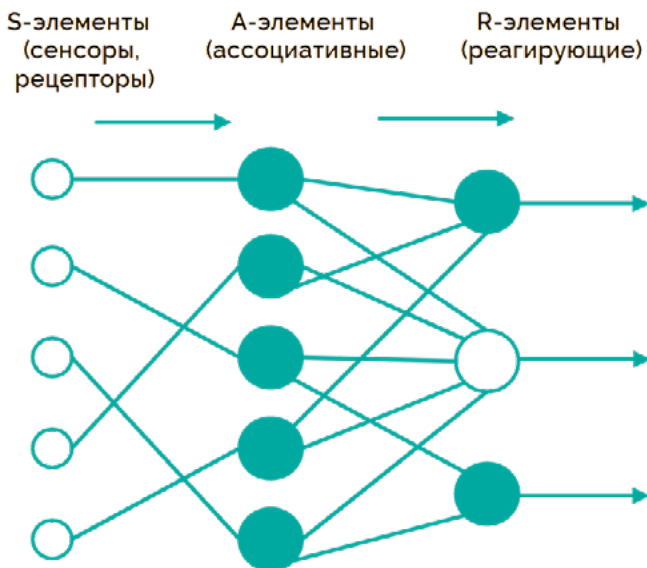
Обучение персептронов, разумно начать с простого однослойного персептрона с одним скрытым слоем. Такая форма позволит работать только с целыми числами.

Простой персептрон

1. A-R связи могут принимать только целые значения (... , -2, -1, 0, 1, 2, ...).
2. У каждого A-элемента может быть только один S-элемент, соответственно и 1 R-элемент.

Преобразуем схему однослойного персептрона под нашу простую задачу.

рис. 8. Однослойный персептрон

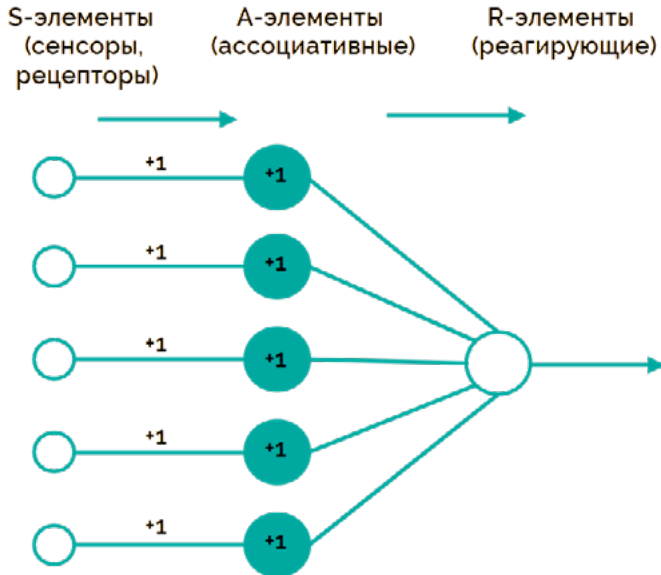


Убираем все лишние связи. A-элементы могут быть соединены только с одним S-элементом и вместо 3 R-элемента оставляем только один.

S-A веса и пороги A элементов у нас теперь равны +1.

Если мы посмотрим внимательно, то станет ясно, что A-элемент не выполняет сейчас никакой функции, так как равен S элементу и его мы можем так же исключить. Итак убираем слой сенсоров и их роль будут выполнять ассоциативные элементы или реагирующие. В данном случае это одно и то же.

рис. 9. Упрощенный персептрон



Вывод: Если A-S связи принимают только целые значения, значит мы можем упростить однослойный персептрон до одного скрытого слоя. В таком упрощенном состоянии нейронная сеть будет продолжать решать задачи на простую классификацию, например распознавание данных.

§ 5.

Обучение персептронов

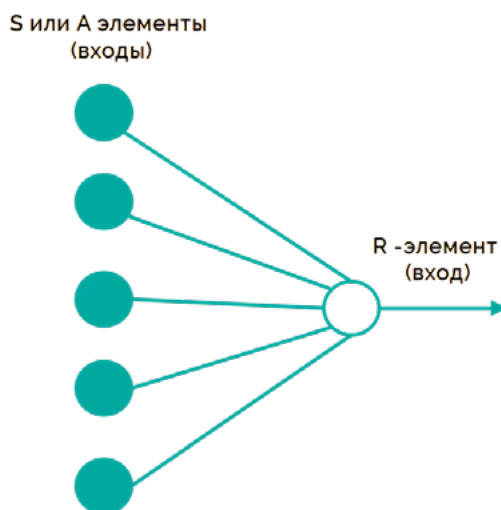
Словарь

- Практика распознавание букв
- Буквы в строковом формате

На прошлом уроке мы упростили модель нейронной сети до простейшего персептрона. (*рис. 9 Упрощенный персептрон*)

Если A-S связи принимают только целые значения, значит мы можем упростить однослойный персептрон до одного скрытого слоя.

рис. 10. S-A элементы персептрона

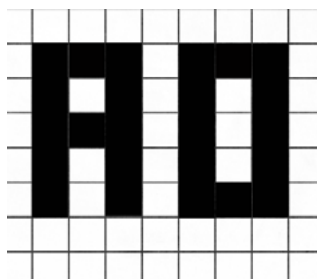


Попробуем научить такую простую модель (к тому же полностью копирующую модель искусственного нейрона) решать простейшую задачу на классификацию. Например, распознавание букв.

Практика: распознавание букв.

Попробуем научить нейросеть распознавать букву А и О.

рис. 12. База данных



Картинка, состоит из пикселей. На каждый пиксель выделим один сенсор (для его распознавания).

То-есть 1 рх-1 сенсор.

- Будем распознавать только черно-белые буквы А и О;
- Цифры будут состоять из черных квадратиков в табличке 3×5 квадратов (Всего 15 сенсоров на каждую букву);
- Нейросеть будет распознавать только одну букву.

В нашей сети будет по 1 S-элементу (он же А-элемент) на каждый квадратик из таблички. Поэтому для распознавания цифры нам потребуется 15 сенсоров. Черный цвет квадрата соответствует возбуждению S-элемента (значение передаваемого сигнала равно 1). Белый цвет – выход соответствующего S-элемента равен 0.

Итак черный цвет-S элемент=1, белый S элемент=0 (так как S и A элементы в нашем упрощенном варианте одно и тоже).

Буквы в строковом формате.

Чтобы работать с нейросетью, мы должны на ее входы подавать сигналы в виде чисел (0 или 1). Таким образом изображение буквы мы должны перевести в последовательность сигналов в виде чисел. Это легко сделать, если представить буквы в строковом формате.

Каждая буква представляет собой всего пятнадцать квадратиков, причем только двух возможных цветов.

За белый квадратик отвечает 0, а черный квадратик – 1.

рис. 13. База данных в строковом формате

	1	1	1		1	1	1
	1	0	1		1	0	1
	1	1	1		1	0	1
	1	0	1		1	0	1
	1	0	1		1	1	1

Для записи каждой буквы у нас используется по 5 строк с 3 символами в каждой. Теперь уберем все переносы строк, чтобы получить для каждой буквы одну длинную строку длиной в 15 символов.

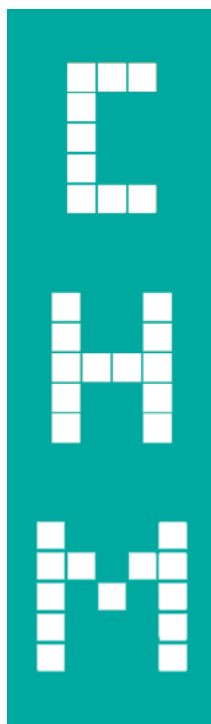
A - 111101111101101

O - 111101101101111

Вывод: Чтобы научить нейросеть распознавать буквы, необходимо использовать сигналы на входе (0,1), каждый сигнал будет обозначать цвет: 0-белый, 1-черный. Картинку с буквами представим в виде пикселей. Для записи каждой буквы используем по 5 строк с 3 символами в каждой. Затем уберем все переносы и запишем одной строкой цепочку в 15 символов по каждой букве. Буквы в таком строковом формате уже можно использовать для работы с нейросетью.

Задание.

1. Укажите верное строковое значение букв С, Н, М в строгом порядке через пробел. Таблица состоит из 5 столбцов и 5 строк для каждой буквы. Все буквы сдвинуты к последнему столбцу, т.е первые столбцы у некоторых букв имеют значение 0. § 5.



§ 6.

Распознавание букв

Словарь

- Постановка задачи
- Алгоритм обучения (подготовка к написанию программы)

Постановка задачи

Мы хотим создать программу, которая из двух букв **А** и **О** будет распознавать нужную нам букву. Например, пусть это будет буква **А**.

Нашей обучающей выборкой будут буквы **А** и **О**.

рис. 14. База данных

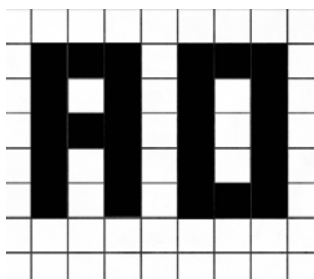


рис. 15. Тестовая выборка. Варианты искаженной буквы А



Для работы нейросети, нам необходимо ряд изображений буквы **А** из обучающей выборки преобразовать в строковый формат.

Таблица для каждой буквы состоит из 5 строк и 5 столбцов, т.е будем использовать 15 сенсоров, запишем в строковом формате без пробелов.

A0 - 111101111101101

A1 - 111100111101101

A2 - 110101110100101

A3 - 110101111100101

Алгоритм обучения

Для обучения нейросети мы будем случайным образом выбирать букву и прогонять ее через сеть, модифицируя ее веса.

Модификация весов – Мы знаем, что важность тем или иным входам (в нашем случае – S-элементам) придают веса, связывающие их с R-элементом. Таким образом, чем сильнее повлиял какой-то вес связи на результат, тем сильнее надо его изменить.

Следовательно, мы должны учесть следующие важные моменты:

- Если наша нейросеть **правильно распознала/отвергла букву A**, то мы ничего не предпринимаем;
- Если нейросеть **ошиблась и распознала неверную букву A как A**, – мы уменьшаем веса тех связей, через которые прошел сигнал. Другими словами веса, связанные с возбуждившимися входами, уменьшаются.
- Если нейросеть **ошиблась и не распознала букву A**, то мы должны увеличить все веса, через которые прошел сигнал.

Таким образом мы как бы говорим сети, что такие связи, а значит и связанные с ними входы – правильные.

Алгоритм обучения, который мы будем реализовывать в программе:

- Подать на входы нейросети букву в строковом формате.
- Если буква распознана/отвергнута верно, то перейти к шагу 1.
- Если сеть ошиблась и распознала неверную букву как A, то вычесть из всех связей, связанных с возбуждившимися S-элементами единицу.
- Если сеть ошиблась и отвергла букву A, то добавить единицу ко всем связям, связанным с возбуждившимися S-элементами.

В алгоритме мы прибавляем или отнимаем именно 1 для нашего удобства, эту величину можно задать любой.

Вывод: Для обучения нейросети, нам необходимо использовать базу данных и выборки. При подаче сигнала в строковом формате на вход нейросеть распознает верную нам букву или отвергает ее, одновременно модифицируя веса. Если нейросеть верно распознала/отвергла букву, то на вход снова подается сигнал, если ошиблась при этом распознала неверную букву, то -1 , если ошиблась и отвергла $+1$ ко всем связям. Таким образом нейросеть учится распознавать буквы.

§ 7.

Обучение нейросети

Программа на языке программирования Python

Словарь

- Программа

Программу для нейросети мы будем делать на языке программирования Python. **Язык программирования Python** универсальный, на сегодняшний момент самый простой в изучении, и является одним из языков программирования допущенных к изучению в школе.

Для работы с языком программирования Python Вы можете использовать онлайн компиляторы, например **РУ** либо загрузить с официального сайта программное обеспечение **Python.org**

Программа

1. Импортируем модуль для работы со случайными числами.

```
import random
```

2. Запишем буквы из обучающей выборки (рис. 14. Обучающая выборка) от А до О в строковом формате без пробелов.

```
# Буквы (Обучающая выборка)
num0 = list('111101111101101')
num1 = list('111101101101111')
```

Обратите внимание, что буква А-0, буква О-1.

3. **Функция list** позволяет нам создать список (массив), состоящий из отдельных символов, на которые разбивается длинная строка. **Добавим буквы А и О в список.**

```
# Список всех букв
nums = [num0, num1]
```

Теперь запишем 3 вида искаженной буквы А в строковом формате. (рис. 15. Тестовая выборка. Варианты искаженной буквы А)

```
# Виды буквы А (Тестовая выборка)
num01 = list('111100111101101')
num02 = list('110101110100101')
num03 = list('110101111100101')
```

4. Создадим список весов. Так как нейронная сеть способна обучиться правильно отвечать из любого состояния, мы смело можем **все веса вначале приравнять 0**. Так как у нас 15 входов и все они сразу соединены с одним R-элементом, то нам потребуется 15 связей. **Напишем инициализацию весов сети.**

```
# Инициализация весов сети
weights = []
for i in range(15):
    weights.append(0)
```

5. Устанавливаем порог. Так как мы используем пороговую функцию активации, то нам необходимо установить какое-то число (порог). Если взвешенная сумма будет больше или равна ему, то сеть выдаст 1 (что означает, что она считает предоставленную букву буквой А). Порог можно выбрать любой. Допустим порог равен 7.

```
# Порог функции активации
bias = 7
```

6. Функция расчета взвешенной суммы она будет считать взвешенную сумму и сравнивать ее с порогом. Фактически эта функция представляет собой единичный шаг работы нашей нейронной сети.

```
# Является ли данная буква буквой А
def proceed(number):

# Рассчитываем взвешенную сумму
    net = 0
    for i in range(15):
        net += int(number[i])*weights[i]

# Превышен ли порог?
    (Да - сеть думает, что это А. Нет - сеть думает, что это другая цифра)
    return net >= bias
```

Обратите внимание, что результатом работы выражения

```
return net >= bias
```

функции может быть True (Правда/Да), что означает 1 или False (Ложь/Нет), что означает 0.

7. Определим еще две вспомогательные функции

Первая функция вызывается, когда сеть считает за букву А неверную букву (выход равен 1 при демонстрации не буква А) и уменьшает на единицу все веса, свя-

занные с возбужденным входами. Напомню, что мы считаем вход возбужденным, если он получил черный пиксель (то есть 1).

```
# Уменьшение значений весов, если сеть ошиблась и выдала 1
def decrease(number):
    for i in range(15):
        # Возбужденный ли вход
        if int(number[i]) == 1:
            # Уменьшаем связанный с ним вес на единицу
            weights[i] -= 1
```

Обратите внимание, на функцию для преобразования символа '1' в цифру 1.

```
int(number[i])
```

Вторая функция вызывается, если сеть не смогла распознать букву А (выход равен 0 при демонстрации буквы А) и увеличивает на единицу все веса, связанные с возбужденными входами.

```
# Увеличение значений весов, если сеть ошиблась и выдала 0
def increase(number):
    for i in range(15):
        # Возбужденный ли вход
        if int(number[i]) == 1:
            # Увеличиваем связанный с ним вес на единицу
            weights[i] += 1
```

8. Обучение нейросети. Делать мы это будем в цикле, который будет повторяться достаточно большое количество раз. Для примера возьмем 10 000 раз. Действия за каждый этап повторения мы описали выше в алгоритме обучения.

```
# Тренировка сети
for i in range(10000):
    # Генерируем случайную букву от 0 (А) до 1 (В)
    option = random.randint(0, 1)
    # Если получилось НЕ буква А (0)
    if option != 0:
        # Если сеть выдала True/Да/1, то наказываем ее
        if proceed(nums[option]):
            decrease(nums[option])
    # Если получилось буква А (0)
    else:
        # Если сеть выдала False/Нет/0, то показываем,
        # что эта буква - то, что нам нужно
```

```
if not proceed(num0):
    increase(num0)
```

9. Выводим результаты обучения.

Сначала выведем результирующие значения весов.

После этого еще раз прогоним сеть по буквам от А до О, но уже без обучения.

Прогоним по тестовой выборке

```
# Вывод значений весов
print(weights)

# Прогон по обучающей выборке
print(«0 это А? », proceed(num0))
print(«1 это А? », proceed(num1))

# Прогон по тестовой выборке
print(«Узнал А? », proceed(num0))
print(«Узнал А - 1? », proceed(num01))
print(«Узнал А - 2? », proceed(num02))
print(«Узнал А - 3? », proceed(num03))
```

10. Результаты.

Вывод: Результаты обучения у каждого могут быть свои, у кого то с третьего раза, у кого-то с 100 раза. Результатом обучения будет способность нейросети распознавать букву А во всех случаях.

```
[1, 1, 1, 1, 0, 1, 1, 6, 1, 1, 0, 1, 1, -5, 1]
0 это А? True
1 это А? False
Узнал А? True
Узнал А - 1? True
Узнал А - 2? True
Узнал А - 3? True
```

Полный код программы:

```
import random

# Буквы (Обучающая выборка)
num0 = list('111101111101101')
num1 = list('111101101101111')
```

```

# Список всех букв
nums = [num0, num1]

# Виды буквы А (Тестовая выборка)
num01 = list('111100111101101')
num02 = list('110101110100101')
num03 = list('110101111100101')

# Инициализация весов сети
weights = []
for i in range(15):
    weights.append(0)

# Порог функции активации
bias = 7

# Является ли данная буква буквой А
def proceed(number):
    # Рассчитываем взвешенную сумму
    net = 0
    for i in range(15):
        net += int(number[i])*weights[i]

    # Превышен ли порог? (Да - сеть думает, что это А. Нет - сеть думает,
    # что это другая цифра)
    return net >= bias

# Уменьшение значений весов, если сеть ошиблась и выдала 1
def decrease(number):
    for i in range(15):
        # Возбужденный ли вход
        if int(number[i]) == 1:
            # Уменьшаем связанный с ним вес на единицу
            weights[i] -= 1

# Увеличение значений весов, если сеть ошиблась и выдала 0
def increase(number):
    for i in range(15):
        # Возбужденный ли вход
        if int(number[i]) == 1:
            # Увеличиваем связанный с ним вес на единицу
            weights[i] += 1

# Тренировка сети

```



```

for i in range(10000):
    # Генерируем случайную букву от 0 (A) до 1 (B)
    option = random.randint(0, 1)
    # Если получилось НЕ буква A (0)
    if option != 0:
        # Если сеть выдала True/Да/1, то наказываем ее
        if proceed(nums[option]):
            decrease(nums[option])
    # Если получилось буква A (0)
    else:
        # Если сеть выдала False/Нет/0, то показываем, что эта буква - то,
        что нам нужно
        if not proceed(num0):
            increase(num0)

# Вывод значений весов
print(weights)

# Прогон по обучающей выборке
print(«0 это A? », proceed(num0))
print(«1 это A? », proceed(num1))

# Прогон по тестовой выборке
print(«Узнал A? », proceed(num0))
print(«Узнал A - 1? », proceed(num01))
print(«Узнал A - 2? », proceed(num02))
print(«Узнал A - 3? », proceed(num03))

```

Практикум

Словарь

- Пишем программу
- Обучаем нейросеть
- Тестируем нейросеть

Задание:

Необходимо написать программу для распознавания буквы О.

рис. 16. Обучающая выборка, база данных

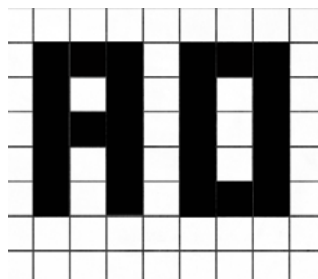


рис. 17. Тестовая выборка



Критерии оценивания задания:

- 0-4 баллов-нейросеть не делает перебор, не обучается;
- 5-7 баллов- нейросеть делает перебор, выводит ошибочные результаты.4
- 7-10 баллов – нейросеть делает перебор выводит правильные.

Заключение

Во 2 части учебного пособия «Искусственные нейронные сети для школьников» «Обучение нейросетей. Профессия нейропрограммист», мы узнали как выглядят простейшие нейросети, персептроны. Познакомились с типами персептронов. Научились обучать простейший однослойный персептрон. Разобрали как нужно писать программы для обучения нейросетей на языке программирования Python, на примере задачи «Распознавания букв».

В следующей 3 части учебного пособия «Искусственные нейронные сети для школьников» «Обучение и тестирование нейросети, профессия тестировщик» – Вы узнаете, что нейросети умеют не только классифицировать в соответствии с выборками, но и решать нестандартные задачи. Вы познакомитесь с правилом Хебба, Дельта-правилом с программированием и обучением нейросети на линейную аппроксимацию. Напишите программу для обучения нейросети на языке программирования Python, протестируете ее.

Официальный сайт учебного пособия

«Искусственные нейронные сети для школьников» proneyroset.ru

Литература

Рышард Тадуесевич, Барбара Боровик, Томаш Гончаж, Бартошц Леппер Элементарное введение в технологию Нейронных сетей с примерами программ // электронная книга

Тарик Рашид. Создаем нейронную сеть // электронная книга.

Полезные ссылки

Нейронные сети для начинающих. Решение задачи классификации Ирисов Фишера // статья [url: <https://habr.com/ru/company/ruvds/blog/679988/>]

Профессиональный информационно-аналитический ресурс, посвященный машинному обучению, распознаванию образов и интеллектуальному анализу данных // электронный ресурс // веб. ресурс machinelearning.ru

Матвеева С.С. Искусственные нейронные сети для школьников // электронное учебное пособие // веб ресурс. proneyroset.ru

Содержание

II Часть

Обучение нейронных сетей, профессия нейропрограммист.

§ 1. История персептрона	4
§ 2. Персептрон	6
§ 3. Классификация персептрона	9
§ 4. Обучение персептронов	13
§ 5. Практика распознавания букв	15
§ 6. Распознавание букв	18
§ 7. Обучение нейросети	20
Практикум	26
Заключение	27

