Министерство Науки и Высшего образования Российской Федерации

Научно-исследовательский технологический университет «МИСиС»

Институт ИТАСУ

Кафедра ЭИИС

Проект **«Распознавание речи»**

по дисциплине

«Интеллектуальные системы и технологии»

Выполнил: Тютиков Д. Л.

Проверил: Танцов П. Н.

Москва 2019

# Оглавление

[Оглавление 2](#_Toc24106297)

[1. Постановка задачи 3](#_Toc24106298)

[2. Исходные данные 4](#_Toc24106299)

[2.1. Создание выборки 4](#_Toc24106300)

[2.2. Расширение выборки программным способом 5](#_Toc24106301)

[2.3. Понижение размерности 6](#_Toc24106302)

[3. Работа с моделями МО 8](#_Toc24106303)

[3.1. Логистическая регрессия, «наивный Баес» и метод N-ближайших соседей 8](#_Toc24106304)

[3.2. Деревья принятия решения 10](#_Toc24106305)

[Нейронные сети 13](#_Toc24106306)

[Нейронная сеть с одним скрытым слоем без нормализации 13](#_Toc24106307)

[Нейронная сеть с одним скрытым слоем и отключением нейронов без нормализации 14](#_Toc24106308)

[Нейронная сеть с несколькими скрытыми слоями без нормализации 15](#_Toc24106309)

[Нейронная сеть с одним скрытым слоем после нормализации 16](#_Toc24106310)

[Нейронная сеть с одним скрытым слоем и отключением нейронов после нормализации 17](#_Toc24106311)

[Нейронная сеть с несколькими скрытыми слоями после нормализации 17](#_Toc24106312)

[Вывод 18](#_Toc24106313)

# 1. Постановка задачи

В ходе выполнения данной проектной работы необходимо создать и обучить систему по распознаванию речи.

Для этого необходимо:

* создать датасет
* снизить размерность
* выбрать и обучить алгоритм МО

Более подробная схема выполнения проекта приведена на рисунке ниже

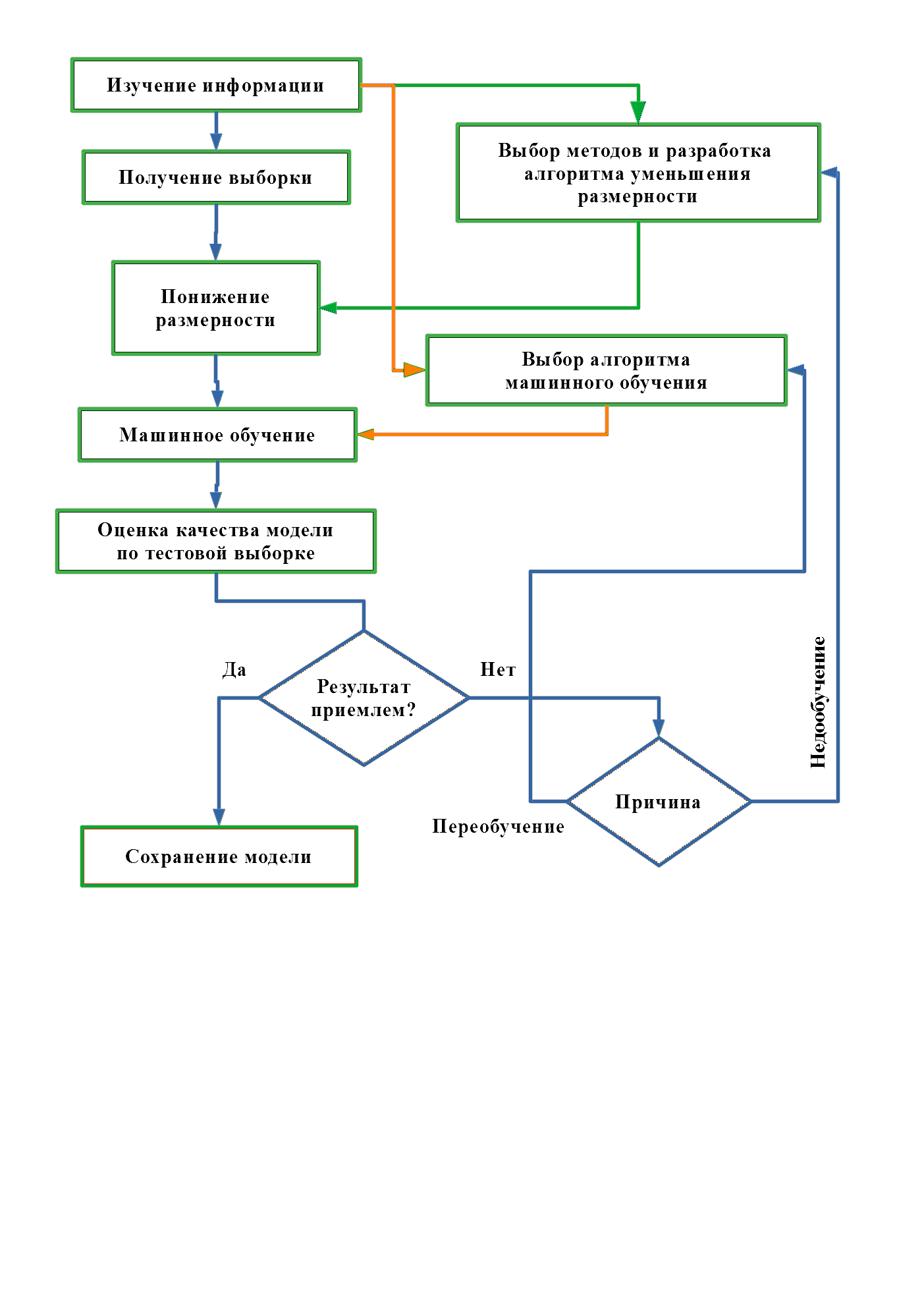


Рис. 1.1. Схема выполнения проекта

Все пункты, представленные на схеме выше, более подробно будут рассмотрены в дальнейшем.

# 2. Исходные данные

## 2.1. Создание выборки

В качестве исходных данных мною была выбрана небольшая коллекция произведений Михаила Жванецкого в исполнении автора. Данные аудиозаписи были представлены в формате mp3, вследствие чего была произведена конвертация при помощи стороннего ресурса в формат wav.

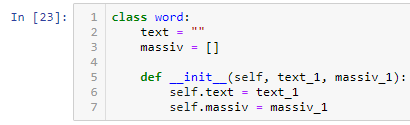
Адрес ресурса: <https://convertio.co/ru/mp3-wav/>

Реализация данной функции имеется и в дополнительно подключаемых библиотеках python, но в этом не было необходимости.

Теперь, когда все данные в читаемом формате, мы можем перейти к созданию выборки из слов. Для нарезки данных аудиопотоков был создан специальный метод, который разбивает поток на сегменты при условии, что звук по амплитуде в течение определенного промежутка времени меньше минимально установленного значения. Этот метод хорошо разбивает поток на отдельные предложения, однако недостаточно гибок для разбиения на слова.

Данный метод по итогу не использовался при создании выборки, однако, он будет представлен в приложении.

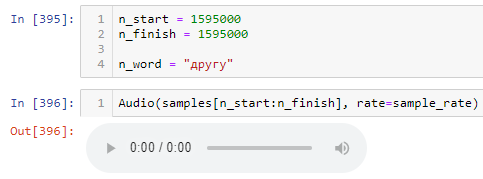
Все разбиение в итоге производилось вручную при помощи методов языка python. Сохранение информации о слове происходит при помощи класса, представленного ниже. Поле text – текст слова. Поле massiv хранит массив значений из потока слова.



n\_start, n\_finish – индекс начала и индекс конца потока, между которыми хранится слово.

n\_word – слово.

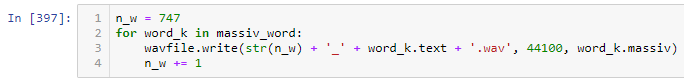
Метод Audio позволяет отобразить значения потока в виде звука. Для выбора слова данные параметры подстраиваются. Проверка выбора осуществляется при



Когда положение слова определено, оно добавляется в массив.



Если все слова записаны в массив, можно их записывать в виде файлов. Для этого используется метод write из класса wavfile из библиотеки scipy.io.



Все слова по итогу записаны отдельных файлах в формате «номер»\_«слово».wav. При помощи номеров в дальнейшем можно будет использовать помимо методов машинного обучения с учителем также и рекуррентные методы, но об этом позже.

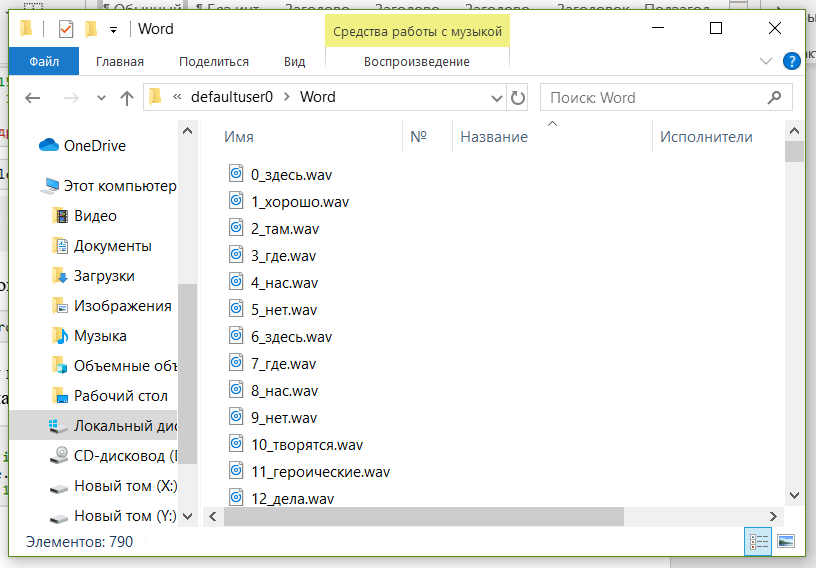


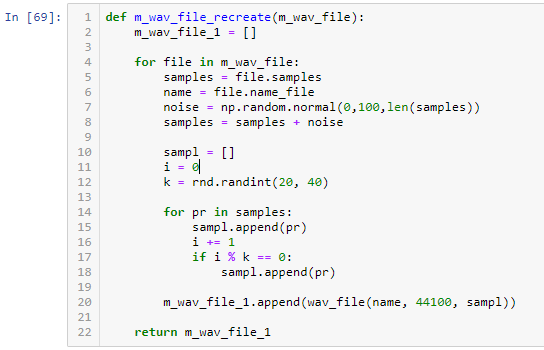
Рис. 2.1. Слова в виде wav-файлов

## 2.2. Расширение выборки программным способом

В данной выборке есть одна принципиальная проблема – ее малый размер. Особенно это принципиально, учитывая тот факт, что многие из элементов-слов имеются в единственном экземпляре. Для увеличения размера выборки программными методами можно использовать следующее:

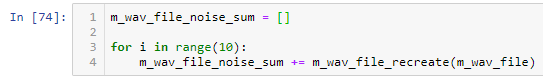
* добавление шума
* дублирование (удаление) определенных значений для изменения АЧХ
* изменение размера слова (добавление в начале и в конце зашумленных полей без информации)

Реализуем данные идеи при помощи следующего алгоритма



Метод random.normal из библиотеки numpy генерирует одномерный массив, размерность которого совпадает с длиной вырезанного сегмента потока слова. Данный метод создает numpy-массив заполненный случайными значениями с нормальным законом распределения.

Затем через случайное количество символов (от 20 до 40) происходит дублирование символа. Сгенерируем при помощи данного метода новые значения выборки.



Это и будет обучающая выборка. Конечно, лучше использовать большое количество слов из разных источников, но имеем, что имеем.

## 2.3. Понижение размерности

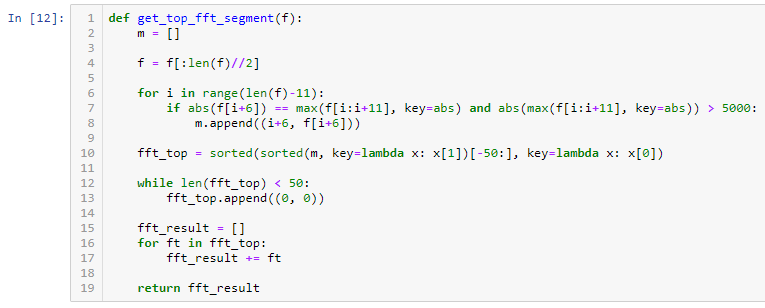
К сожалению, все слова, представленные в выборках, даже без учета преобразований, представленных ранее, имеют различную размерность. Насколько мне известно, имеются реализации методов МО, использующие признаки переменной длины, однако при данной длине вектора признаков их использовать не целесообразно. К тому же количество признаков для корректной работы в условии имеющихся мощностей слишком велико.

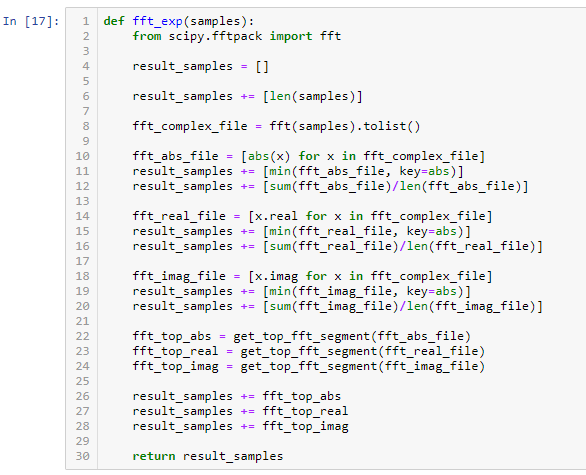
Для каждого сигнала можно выделить следующие характеристики:

* Длина слова в потоке
* АЧХ (по модулю):
  + Значение пиковых значений АЧХ
  + Положение пиковых значений относительно частоты
  + Среднее значение
  + Медианное значение
* АЧХ (действительная часть):
  + Значение пиковых значений АЧХ
  + Положение пиковых значений относительно частоты
  + Среднее значение
* АЧХ (мнимая часть):
  + Значение пиковых значений АЧХ
  + Положение пиковых значений относительно частоты
  + Среднее значение

Как видно, в качестве «цензора» в нашем случае будет использоваться амплитудно-частотная характеристика. Это вызвано тем, что звук представляет из себя совокупность гармонических колебаний, которые проще оценивать при помощи АЧХ. Также выборка по пиковым значениям снижает влияние шума.

Для получения спектра сигнала используется метод fft из библиотеки scipy. Данный метод получает гармоники спектра сигнала в комплексном виде.





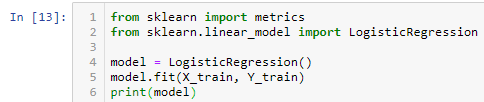
# 3. Работа с моделями МО

Прежде чем приступить к обучению, создадим два тестовых набора: первый представляет из себя исходный звук, пропущенный через метод fft\_exp, второй – зашумленный, аналогично обучающей выборке.

Для обучения будет использоваться библиотека scikit-learn. Данная библиотека представляет обширный стек оптимизированных методов МО. Также в данной библиотеке представлены методы по нормализации и методы вывода информации о результатах обучения: последние расположены в классе metrics.

## 3.1. Логистическая регрессия, «наивный Баес» и метод N-ближайших соседей

Код для обучения модели логистической регрессии приведен ниже. Он практически идентичен для всех используемых моделей.



В данном пункте также использованы:

- GaussianNB – наивный баесовский классификатор (naive\_bayes)

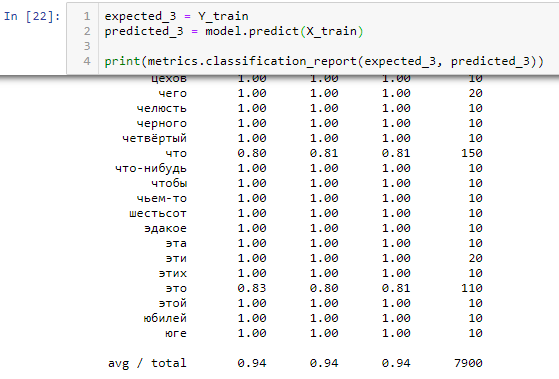
- KNeighborsClassifier – метод N-ближайших соседей (neighbors)

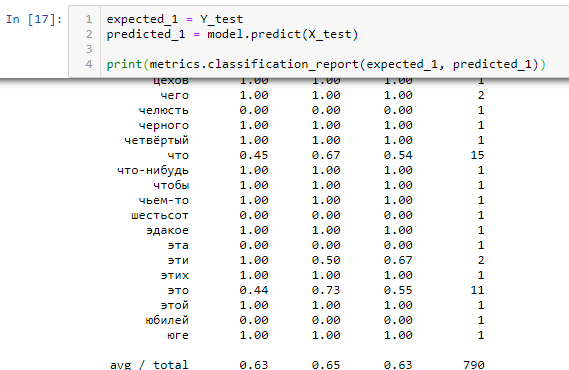
Логистическая регрессия – это статистическая модель, используемая для прогнозирования вероятности возникновения некоторого события путём подгонки данных к логистической кривой. Логистическая регрессия применяется для прогнозирования вероятности возникновения некоторого события по значениям множества признаков.

Байесовский классификатор – простой вероятностный классификатор, основанный на применении теоремы Байеса со строгими (наивными) предположениями о независимости. В связи с данным допущением получается, что вероятность того, что элемент принадлежит тому или иному классу прямо пропорционально произведению вероятностей признаков: вероятность того, что данное значение признака определяет данное значение класса.

Метод N-ближайших соседей: определяются ближайшие по признакам элементы из обучающей выборки, после чего элементу присваивается тот класс, к которому относится большее количество соседних элементов.

Реализация получения предсказаний и оценки точности модели по данным предсказаниям представлено ниже.





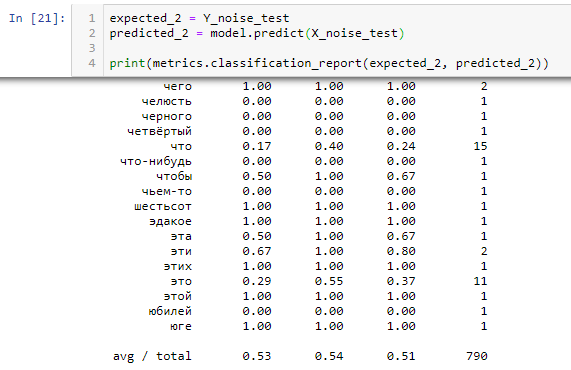


Таблица 1. Оценка качества работы обученных моделей

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Логистическая регрессия | «Наивный Баес» | Метод N-ближайших соседей | | |
| 3 | 7 | 11 |
| Проверка на обучающей выборке | | | | | |
| Точность | 0.94 | 0.81 | 0.94 | 0.88 | 0.83 |
| Полнота | 0.94 | 0.78 | 0.92 | 0.83 | 0.76 |
| F1 – оценка | 0.94 | 0.77 | 0.92 | 0.83 | 0.76 |
| Без шума | | | | | |
| Точность | 0.63 | 0.41 | 0.88 | 0.82 | 0.77 |
| Полнота | 0.65 | 0.46 | 0.88 | 0.81 | 0.74 |
| F1 – оценка | 0.63 | 0.40 | 0.87 | 0.79 | 0.73 |
| С шумом | | | | | |
| Точность | 0.53 | 0.72 | 0.85 | 0.77 | 0.70 |
| Полнота | 0.54 | 0.71 | 0.85 | 0.76 | 0.69 |
| F1 – оценка | 0.51 | 0.69 | 0.83 | 0.74 | 0.66 |
| Время обучения | > 20 минут | 0.5 c. | 0.5 c. | | |

Как можно видеть из таблицы, логистическая регрессия имеет лучшие показатели по обучающей выборке и одни из худших по тестовым, что, скорее всего, свидетельствует о переобучении.

Также стоит заметить, что по методу Баеса распознавание слов без шума производится менее качественно, чем распознавание с шумом. Переобучение в данном случае не значительно, однако такая реакция на отсутствие шума говорит о том, что использование данного метода для представленных данных не совсем уместно.

Метод N-ближайших соседей, являясь наиболее простым, показал самую высокую эффективность без учета влияния шума. Интересно, что увеличение соседей, по которым ведется оценка снижает не только скорость получения прогнозируемых классов, но и точность.

## 3.2. Деревья принятия решения

Теперь остановимся на древовидных структурах принятия решений.

Первым делом рассмотрим базовую структуру: дерево принятия решения. Практически любая модель любого типа имеет гиперпараметры. В предыдущем пункте таким гиперпараметром было количество соседей. В данном – глубина, на которой происходит обучение.

Таблица 2. Оценка качества работы деревьев принятия решения в зависимости от максимальной глубины

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Дерево принятия решения | | | | |
| 5 | 10 | 15 | 20 | None |
| Проверка на обучающей выборке | | | | | |
| Точность | 0.11 | 0.36 | 0.63 | 0.80 | 1.00 |
| Полнота | 0.12 | 0.33 | 0.58 | 0.73 | 1.00 |
| F1 – оценка | 0.08 | 0.31 | 0.57 | 0.74 | 1.00 |
| Без шума | | | | | |
| Точность | 0.08 | 0.22 | 0.41 | 0.49 | 0.64 |
| Полнота | 0.11 | 0.21 | 0.37 | 0.47 | 0.64 |
| F1 – оценка | 0.07 | 0.19 | 0.35 | 0.45 | 0.62 |
| С шумом | | | | | |
| Точность | 0.11 | 0.30 | 0.51 | 0.65 | 0.84 |
| Полнота | 0.12 | 0.30 | 0.49 | 0.63 | 0.85 |
| F1 – оценка | 0.08 | 0.27 | 0.47 | 0.62 | 0.83 |
| Время обучения | 8.36 с. | 16.18 с. | 22.71 с. | 25.86 с. | 31.79 |

Для снижения уровня переобучения вместо обычного дерева может применяться случайный. Если алгоритм первого включает разбиение при получении минимального значения функции ошибки: минимальное количество элементов классов должно быть отнесены к другим классам, то алгоритм случайного дерева производит разбиение случайным образом, присваивая тот класс, который встречается в данном сегменте наиболее часто.

Таблица 3. Оценка качества работы случайных деревьев   
принятия решения в зависимости от максимальной глубины

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Дерево принятия решения | | | | |
| 5 | 10 | 15 | 20 | None |
| Проверка на обучающей выборке | | | | | |
| Точность | 0.07 | 0.31 | 0.54 | 0.70 | 0.87 |
| Полнота | 0.10 | 0.26 | 0.48 | 0.63 | 0.86 |
| F1 – оценка | 0.06 | 0.24 | 0.48 | 0.63 | 0.86 |
| Без шума | | | | | |
| Точность | 0.05 | 0.20 | 0.39 | 0.42 | 0.63 |
| Полнота | 0.09 | 0.21 | 0.38 | 0.42 | 0.64 |
| F1 – оценка | 0.05 | 0.17 | 0.36 | 0.39 | 0.61 |
| С шумом | | | | | |
| Точность | 0.05 | 0.21 | 0.38 | 0.45 | 0.64 |
| Полнота | 0.08 | 0.21 | 0.38 | 0.45 | 0.66 |
| F1 – оценка | 0.05 | 0.18 | 0.35 | 0.43 | 0.63 |
| Время обучения | 0.86 с. | 1.01 с. | 1.31 с. | 1.33 с. | 1.5 |

Как можно заметить, данный метод выполняется гораздо быстрее аналога. «Случайное дерево» снижает переобучение от вносимого шума – как можно видеть результаты на тестовой выборке без шума и с шумом практически одинаковы. С другой стороны, обыкновенное «дерево принятия решения» имело лучший результат, как по исходной, так и по зашумленной тестовой выборке.

Для повышения эффективности часто используется метод «случайного леса»: в данном алгоритме производится обучение ансамбля «случайных деревьев». После обучения каждое дерево по каждому из элементов, относя его к тому или иному классу. По итогу выбирается тот класс, за который проголосовало большее число деревьев.

Таблица 4. Оценка качества работы случайного леса в зависимости от   
количества деревьев

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Случайный лес | | | | |
| 2 | 3 | 5 | 7 | 10 |
| Проверка на обучающей выборке | | | | | |
| Точность | 0.88 | 0.96 | 0.99 | 1.00 | 1.00 |
| Полнота | 0.87 | 0.96 | 0.99 | 1.00 | 1.00 |
| F1 – оценка | 0.87 | 0.96 | 0.99 | 1.00 | 1.00 |
| Без шума | | | | | |
| Точность | 0.61 | 0.75 | 0.86 | 0.90 | 0.93 |
| Полнота | 0.63 | 0.75 | 0.84 | 0.88 | 0.90 |
| F1 – оценка | 0.58 | 0.73 | 0.83 | 0.88 | 0.90 |
| С шумом | | | | | |
| Точность | 0.64 | 0.73 | 0.91 | 0.92 | 0.96 |
| Полнота | 0.66 | 0.75 | 0.91 | 0.92 | 0.96 |
| F1 – оценка | 0.62 | 0.72 | 0.90 | 0.91 | 0.96 |
| Время обучения | 3.07 с. | 4.54 с. | 6.97 с. | 9.79 с. | 13.10 с. |

По полученным результатам можно видеть, что случайный лес является более быстрым и эффективным способом машинного обучения, чем отдельные деревья решений. Также данный метод в меньшей степени подвержен переобучению.

## Нейронные сети

Теперь перейдем к искусственным нейронным сетям. Данный метод подразумевает построение однослойного или многослойного персептрона. Для реализации искуственной нейронной сети используется библиотека keras.

В ходе обучения использованы следующие параметры:

– оптимизатор: ADAM;

– функция ошибки: кросс-энтропия;

– количество эпох: 4;

– функция активации: софтмакс;

Данные параметры используются, если не указано другое.

Для работы с нейронной сетью необходимо представить вектор слов-классов в виде матрицы смежности.

### Нейронная сеть с одним скрытым слоем без нормализации

Первым будет использована модель с одним скрытым слоем. В качестве изменяющегося гиперпараметра в данном случае будем использовать количество нейронов в скрытом слое. На рисунке ниже представлена упрощенная схема данного алгоритма.

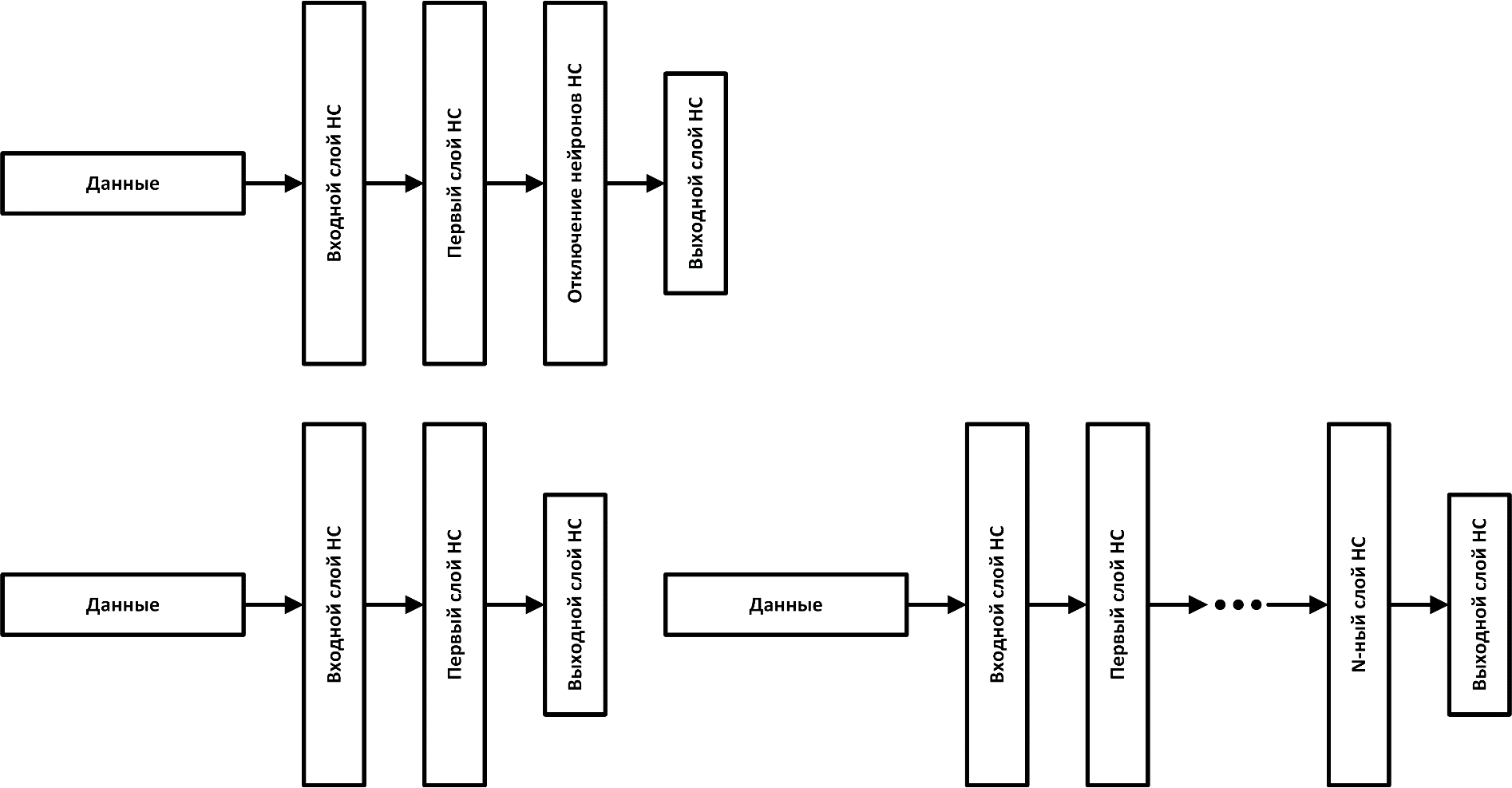


Рис. 1. Структура алгоритма НС с одним  
скрытым слоем

Таблица 5. Оценка качества работы нейронной сети с одним  
внутренним слоем

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Кол-во нейронов | 5 | 10 | 20 | 40 | 80 | 160 |
| Проверка на обучающей выборке | | | | | | |
| Точность | 0.53 | 0.57 | 0.66 | 0.76 | 0.79 | 0.76 |
| Без шума | | | | | | |
| Точность | 0.53 | 0.55 | 0.60 | 0.72 | 0.72 | 0.71 |
| С шумом | | | | | | |
| Точность | 0.53 | 0.56 | 0.59 | 0.70 | 0.71 | 0.71 |
| Время обучения | 10 c. | 10 c. | 10 c. | 10 c. | 10 c. | 10 c. |

Стоит отметить, что веса в нейронной сети устанавливаются случайным образом, в связи с чем разброс результатов после обучения достаточно высок.

### Нейронная сеть с одним скрытым слоем и отключением нейронов без нормализации

В прошлом пункте можно заметить небольшое переобучение. Чтобы снизить уровень переобучения можно использовать отключение нейронов. Отключение нейронов проводится случайным образом для каждой эпохи.

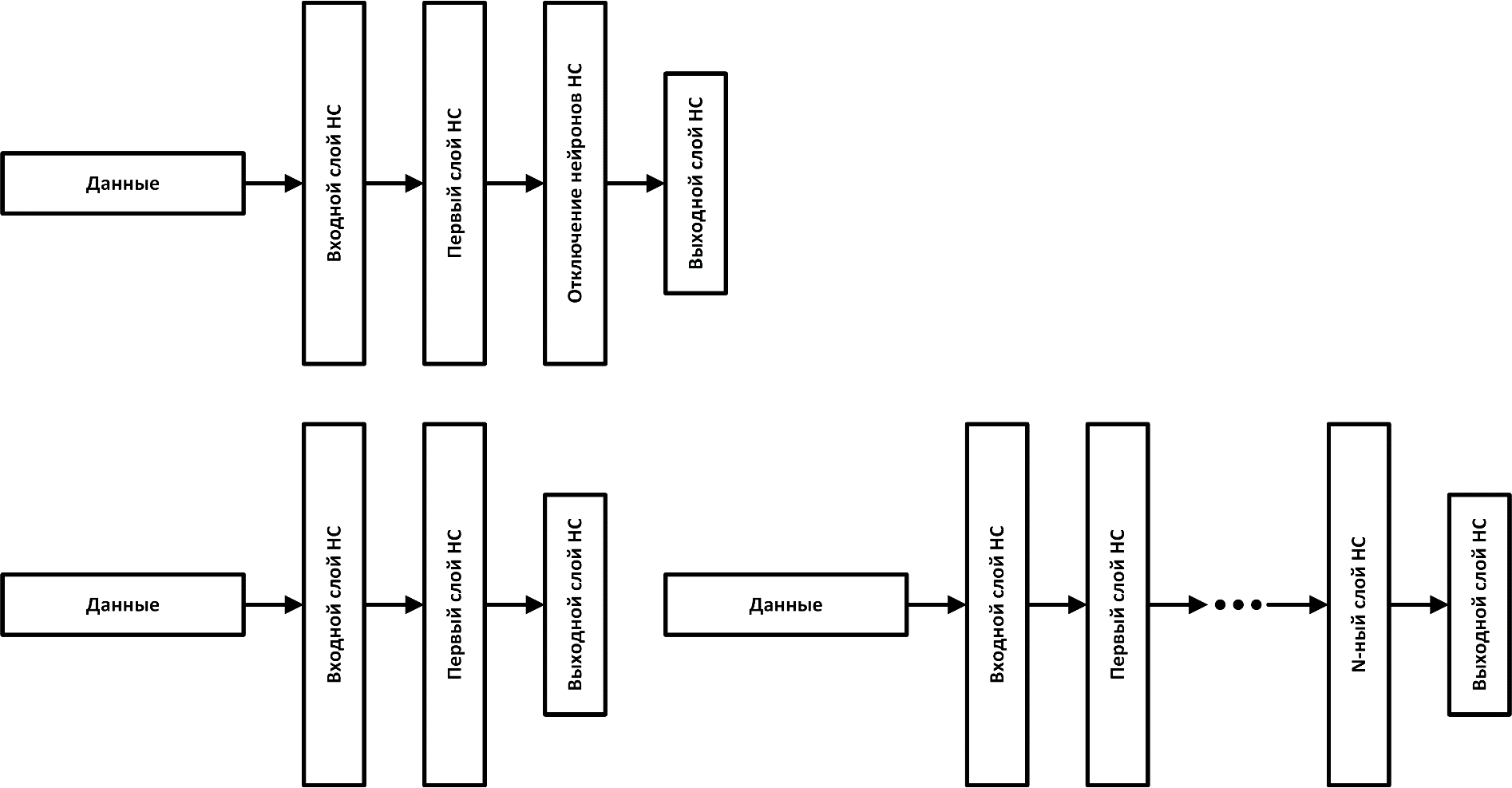


Рис. 2. Структура алгоритма НС с одним  
скрытым слоем и отключением нейронов

Таблица 6. Оценка качества работы нейронной сети с одним  
внутренним слоем и отключением нейронов

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Кол-во нейронов | 20 k=0.25 | 40 k=0.25 | 80 k=0.25 | 20 k=0.5 | 40 k=0.5 | 80 k=0.5 | 20 k=0.75 | 40 k=0.75 | 80 k=0.75 |
| Проверка на обучающей выборке | | | | | | | | | |
| Точность | 0.54 | 0.70 | 0.76 | 0.62 | 0.62 | 0.76 | 0.63 | 0.67 | 0.66 |
| Без шума | | | | | | | | | |
| Точность | 0.54 | 0.67 | 0.73 | 0.6 | 0.61 | 0.72 | 0.65 | 0.65 | 0.63 |
| С шумом | | | | | | | | | |
| Точность | 0.53 | 0.66 | 0.72 | 0.59 | 0.61 | 0.73 | 0.62 | 0.65 | 0.65 |
| Время обучения | 12 c. | 15 c. | 16 c. | 15 c. | 16 c. | 16 c. | 16 c. | 17 c. | 16 c. |

Точность стала меньше по сравнению с предыдущим пунктом. Результаты стали более случайны, что объясняется случайным отключением нейронов.

### Нейронная сеть с несколькими скрытыми слоями без нормализации

От сетей с одним скрытым слоем перейдем к оценке эффективности сетей с несколькими скрытыми слоями.

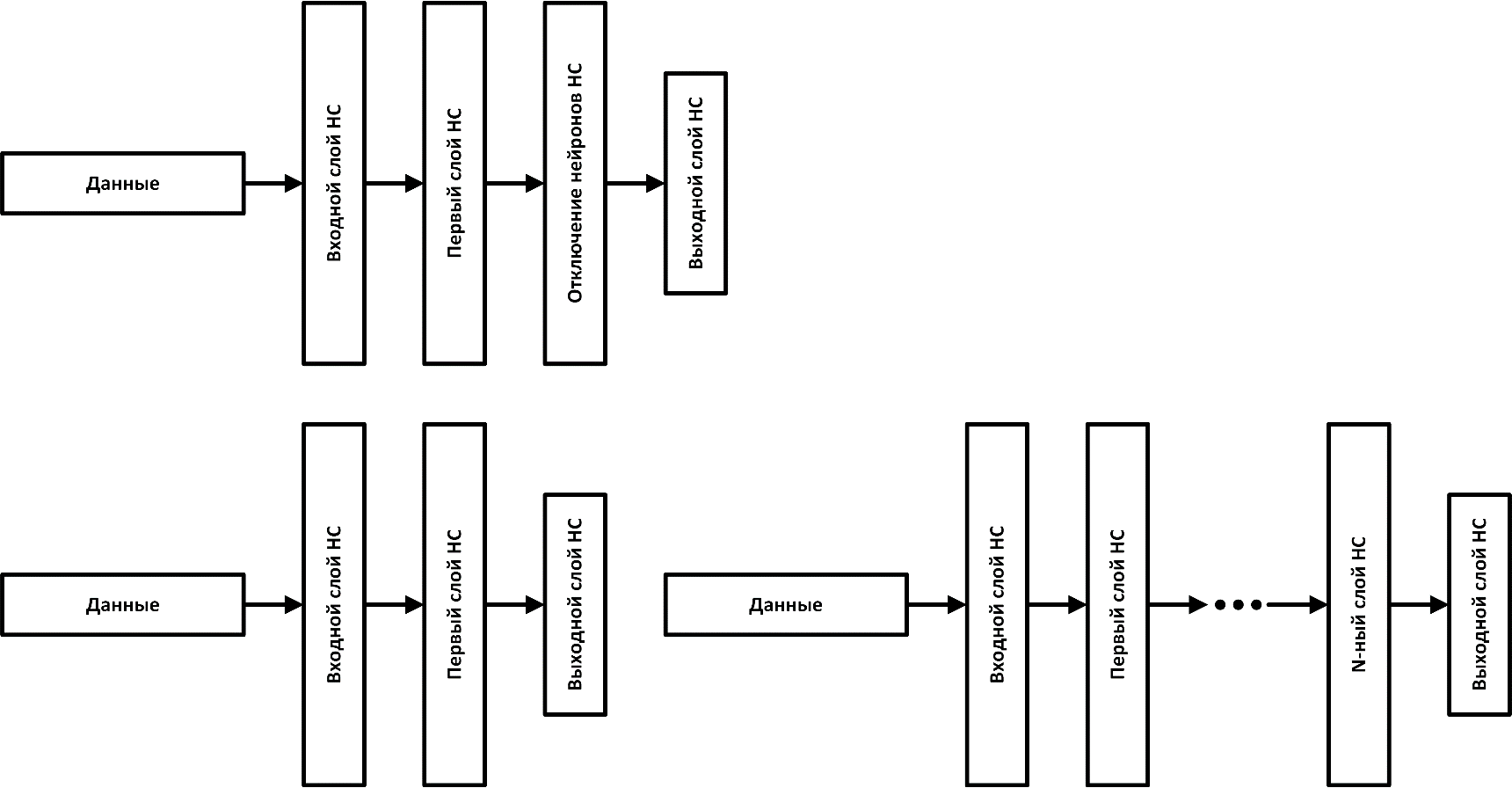


Рис. 3. Структура алгоритма многослойной НС

Таблица 7. Оценка качества работы нейронной  
сети с несколькими скрытыми слоями

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Кол-во слоев | 2 | 2 | 2 | 3 | 3 | 3 | 3 | 4 | 5 |
| Размер слоя 1 | 20 | 40 | 80 | 20 | 40 | 80 | 5 | 20 | 80 |
| Размер слоя 2 | 20 | 20 | 40 | 20 | 20 | 40 | 5 | 20 | 40 |
| Размер слоя 3/4/5 | – | – | – | 10 | 10 | 20 | 5 | 20 | 20 |
| Проверка на обучающей выборке | | | | | | | | | |
| Точность | 0.58 | 0.59 | 0.55 | 0.51 | 0.51 | 0.52 | 0.51 | 0.5 | 0.49 |
| Без шума | | | | | | | | | |
| Точность | 0.58 | 0.60 | 0.54 | 0.50 | 0.51 | 0.52 | 0.50 | 0.51 | 0.48 |
| С шумом | | | | | | | | | |
| Точность | 0.58 | 0.58 | 0.55 | 0.50 | 0.51 | 0.52 | 0.50 | 0.50 | 0.49 |
| Время обучения | 11 с. | 12 c. | 16 c. | 16 c. | 15 c. | 20 c. | 17 c. | 20 c. | 21 c. |

Как можно видеть, увеличение количество слоев привело к снижению качества работы нейронной сети. Увеличение количества эпох до 20 привело к незначительному увеличению эффективности (в среднем на 0.5 – 1 %).

### Нейронная сеть с одним скрытым слоем после нормализации

От работы с ненормализованной выборкой перейдем к работе с нормализованной. Нормализация выборки осуществлялась при помощи метода MinMaxScaler из библиотеки scikit learn. Схема алгоритма примет следующий вид.

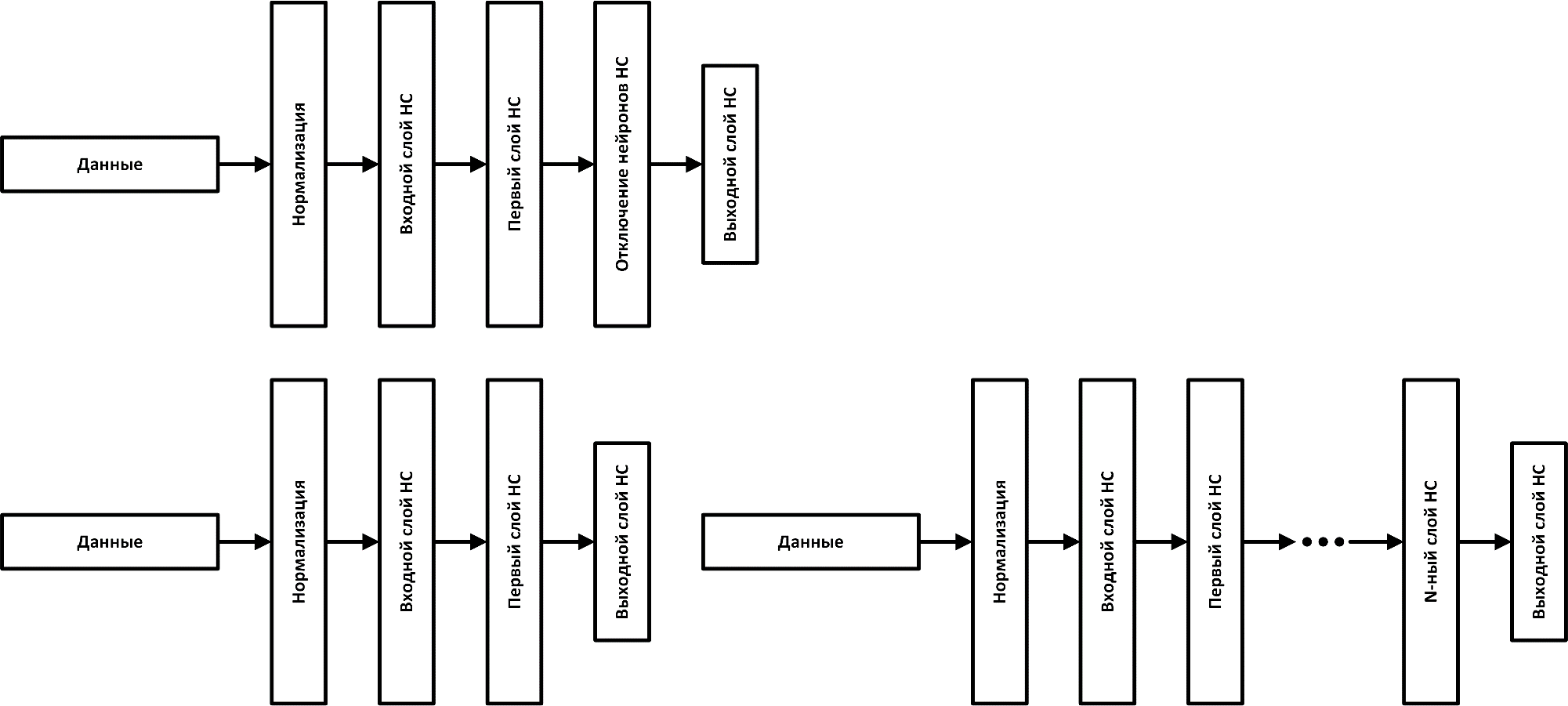


Рис. 4. Структура алгоритма НС с одним  
скрытым слоем

Таблица 5. Оценка качества работы нейронной сети с одним  
внутренним слоем

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Кол-во нейронов | 5 | 10 | 20 | 40 | 80 | 160 |
| Проверка на обучающей выборке | | | | | | |
| Точность | 0.64 | 0.71 | 0.76 | 0.78 | 0.79 | 0.80 |
| Без шума | | | | | | |
| Точность | 0.65 | 0.72 | 0.76 | 0.78 | 0.80 | 0.81 |
| С шумом | | | | | | |
| Точность | 0.65 | 0.71 | 0.76 | 0.78 | 0.79 | 0.80 |
| Время обучения | 4 c. | 4 c. | 4 c. | 4 c. | 4 c. | 4 c. |

Можно отметить, что результаты стали лучше. Также стоит отметить отсутствие переобучения.

### Нейронная сеть с одним скрытым слоем и отключением нейронов после нормализации

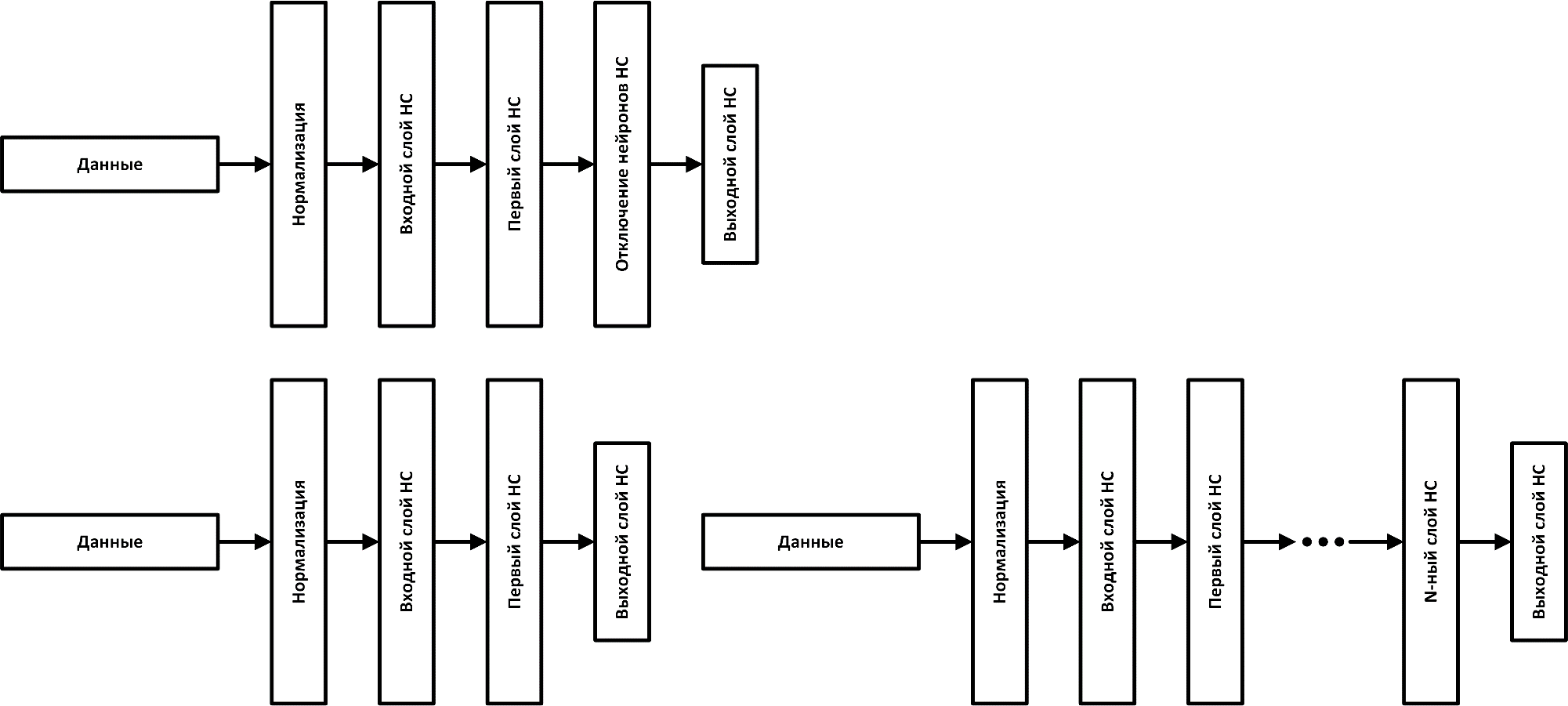


Рис. 5. Структура алгоритма НС с одним  
скрытым слоем и отключением нейронов

Таблица 6. Оценка качества работы нейронной сети с одним  
внутренним слоем и отключением нейронов

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Кол-во нейронов | 20 k=0.25 | 40 k=0.25 | 80 k=0.25 | 20 k=0.5 | 40 k=0.5 | 80 k=0.5 | 20 k=0.75 | 40 k=0.75 | 80 k=0.75 |
| Проверка на обучающей выборке | | | | | | | | | |
| Точность | 0.71 | 0.76 | 0.77 | 0.64 | 0.68 | 0.71 | 0.6 | 0.6 | 0.61 |
| Без шума | | | | | | | | | |
| Точность | 0.7 | 0.76 | 0.77 | 0.64 | 0.68 | 0.71 | 0.6 | 0.59 | 0.6 |
| С шумом | | | | | | | | | |
| Точность | 0.71 | 0.76 | 0.77 | 0.65 | 0.68 | 0.71 | 0.6 | 0.6 | 0.61 |
| Время обучения | 6 c. | 6 c. | 6 c. | 6 c. | 6 c. | 6 c. | 6 c. | 6 c. | 6 c. |

### Нейронная сеть с несколькими скрытыми слоями после нормализации

…

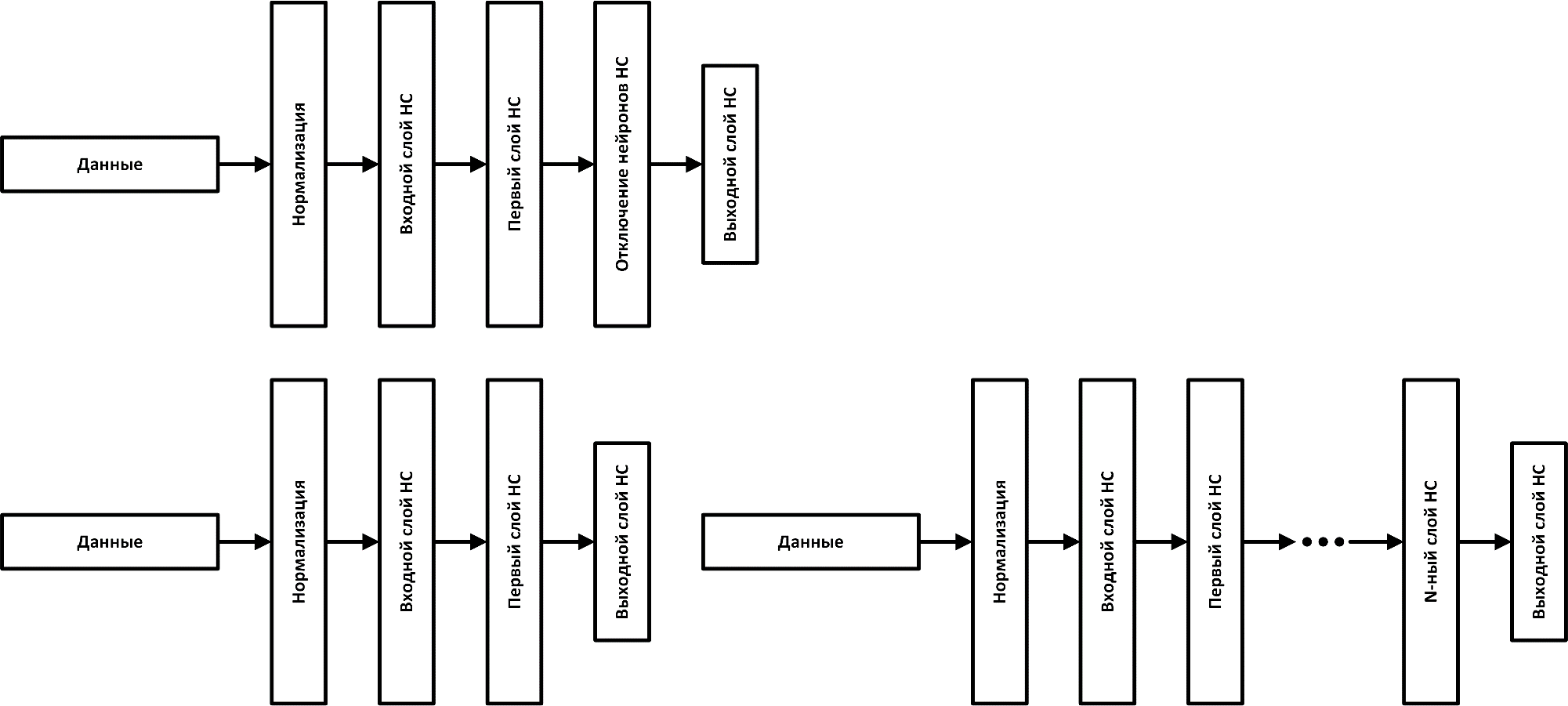


Рис. 6. Структура алгоритма многослойной НС

Таблица 7. Оценка качества работы нейронной  
сети с несколькими скрытыми слоями

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Кол-во слоев | 2 | 2 | 2 | 3 | 3 | 3 | 3 | 4 | 5 |
| Размер слоя 1 | 20 | 40 | 80 | 20 | 40 | 80 | 5 | 20 | 80 |
| Размер слоя 2 | 20 | 20 | 40 | 20 | 20 | 40 | 5 | 20 | 40 |
| Размер слоя 3/4/5 | – | – | – | 10 | 10 | 20 | 5 | 20 | 20 |
| Проверка на обучающей выборке | | | | | | | | | |
| Точность | 0.54 | 0.54 | 0.53 | 0.50 | 0.53 | 0.52 | 0.51 | 0.5 | 0.49 |
| Без шума | | | | | | | | | |
| Точность | 0.54 | 0.54 | 0.53 | 0.50 | 0.51 | 0.52 | 0.52 | 0.51 | 0.48 |
| С шумом | | | | | | | | | |
| Точность | 0.53 | 0.52 | 0.54 | 0.50 | 0.51 | 0.52 | 0.50 | 0.50 | 0.49 |
| Время обучения | 12 с. | 11 c. | 13 c. | 14 c. | 16 c. | 17 c. | 16 c. | 18 c. | 18 c. |

По итогу можно сказать, следующее:

- самой эффективной НС для данного набора данных является сеть с одним скрытым слоем,

- увеличение слоев при обучении по исходному датасету ведет к снижению качества обучения

- отключение нейронов ведет к снижению переобучения, однако также это может привести к снижению качества обучения.

# Вывод

В ходе данной работы был разработан алгоритм понижения размерности для аудиопотока и была проанализирована эффективность различных методов машинного обучения.

Наиболее эффективным методом для данной выборки является «случайный лес».

Данная тематика предполагает более обширную область для работ, чем то, что реализовано и представлено в данном отчете.