ФЕДЕРАЛЬНОЕ АГЕНСТВО СВЯЗИ

Ордена Трудового Красного Знамени федеральное государственное бюджетное образовательное учреждение высшего образования

Московский технический университет связи и информатики

(МТУСИ)

Кафедра «Математическая кибернетика и информационные технологии» Дисциплина «Интеллектуальные системы»

Лабораторная работа № 1

Многоклассовая классификация цветов

Выполнил: \_\_\_\_\_\_\_ Д.С. Калининский

Проверил: \_\_\_\_\_\_\_ А.В. Шереметьев

Москва 2022

**Оглавление**

[1 Цель работы 3](#_Toc119580633)

[2 Задачи 3](#_Toc119580634)

[3 Выполнение работы 3](#_Toc119580635)

[4 Результаты 6](#_Toc119580636)

[5 Вывод 8](#_Toc119580637)

# **1 Цель работы**

Цель лабораторной работы, реализовать классификацию сортов растения ирис (Iris Setosa - 0, Iris Versicolour - 1, Iris Virginica - 2) по четырем признакам: размерам пестиков и тычинок его цветков.

# **2 Задачи**

Основные задачи лабораторной работы:

1) Ознакомиться с задачей классификации;

2) Загрузить данные;

3)Создать модель ИНС;

4)Настроить параметры обучения;

5)Обучить и оценить модель.

# **3 Выполнение работы**

Задача многоклассовой классификации является одним из основных видов задач, для решения которых применяются нейронные сети. На рисунке 1 представлен пример данных.

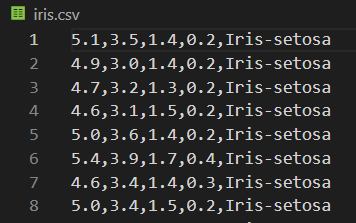


Рисунок 1 – Пример данных

Импортируем необходимые для работы классы и функции. Кроме Keras понадобится Pandas для загрузки данных и scikit-learn для подготовки данных и оценки модели. Подключение модулей import pandas from tensorflow.keras.layers import Dense from tensorflow.keras.models import Sequential from tensorflow.keras.utils import to\_categorical from sklearn.preprocessing import LabelEncoder (рисунок 2).

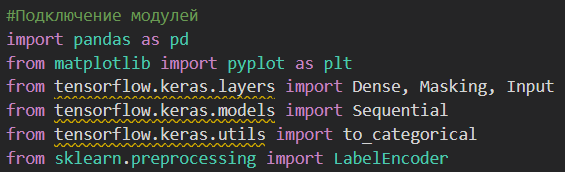


Рисунок 2 – Подключение модулей

Набор данных загружается напрямую с помощью pandas. Затем необходимо разделить атрибуты (столбцы) на входные данные(X) и выходные данные(Y) (рисунок 3).

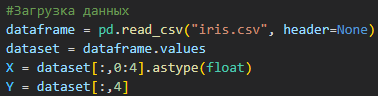


Рисунок 3 – Загрузка данных

При решении задач многоклассовой классификации хорошей практикой является преобразование выходных атрибутов из вектора в матрицу к виду представленных ниже (рисунок 4).

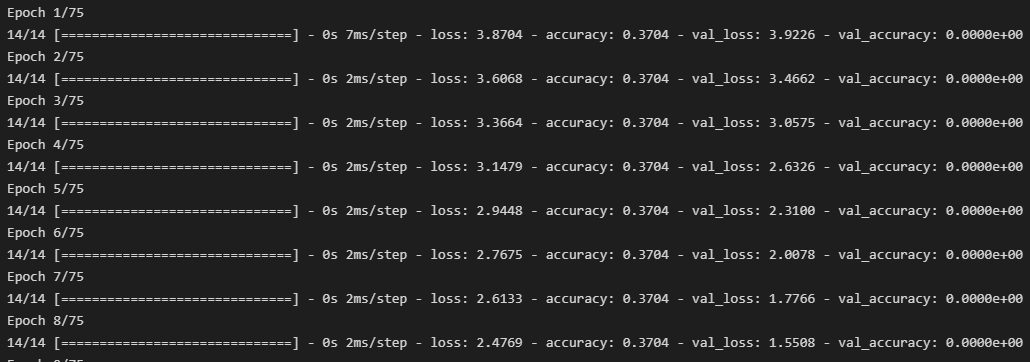


Рисунок 4 - Представление данных

Для этого необходимо использовать функцию to\_categorical().

Переход от текстовых меток к категориальному вектору (рисунок 5).

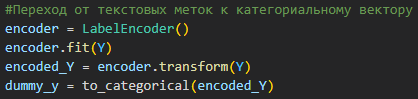


Рисунок 5 – Конвертация данных

Теперь можно задать базовую архитектуру сети (рисунок 6).

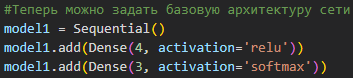


Рисунок 6 – Базовая архитектура сети

Основным строительным блоком нейронных сетей является слой (или уровень), модуль обработки данных, который можно рассматривать как фильтр для данных. Он принимает некоторые данные и выводит их в более полезной форме. В частности, слои извлекают представления из подаваемых в них данных, которые, как мы надеемся, будут иметь больше смысла для решаемой задачи. Фактически методика глубокого обучения заключается в объединении простых слоев, реализующих некоторую форму поэтапной очистки данных. Модель глубокого обучения можно сравнить с ситом, состоящим из последовательности фильтров все более тонкой очистки данных — слоев. В данном случае наша сеть состоит из последовательности двух слоев Dense, которые являются тесно связанными (их еще называют полносвязными) нейронными слоями. Второй (и последний) слой — это 3-переменный слой потерь (softmax layer), возвращающий массив с 3 оценками вероятностей (в сумме дающих 1). Каждая оценка определяет вероятность принадлежности текущего изображения к одному из 3 классов цветов.

Чтобы подготовить сеть к обучению, нужно настроить еще три параметра для этапа компиляции:

1) функцию потерь, которая определяет, как сеть должна оценивать качество своей работы на обучающих данных и, соответственно, как корректировать ее в правильном направлении;

2) оптимизатор — механизм, с помощью которого сеть будет обновлять себя, опираясь на наблюдаемые данные и функцию потерь;

3) метрики для мониторинга на этапах обучения и тестирования — здесь нас будет интересовать только точность (доля правильно классифицированных изображений).

Инициализация параметров обучения (рисунок 7).



Рисунок 7 – Параметры обучения

Теперь можно начинать обучение сети, для чего в случае использования библиотеки Keras достаточно вызвать метод fit сети — он пытается адаптировать (fit) модель под обучающие данные.

Обучение сети (рисунок 8).



Рисунок 8 – Обучение сети

В процессе обучения отображаются четыре величины: потери сети на обучающих данных и точность сети на обучающих данных, а также потери и точность на данных, не участвовавших в обучении.

# **4 Результаты**

В целях изучения различных архитектур ИНС и обучение при разных параметрах обучения, задаём три модели (рисунок 9) с разными количеством слоёв, количеством нейронов на слоях и параметрах функции fit.



Рисунок 9 – Задаём три модели

Для получения визуальных результатов на рисунках 10-12 представлены графики, используем библиотеки matplotlib и numpy.

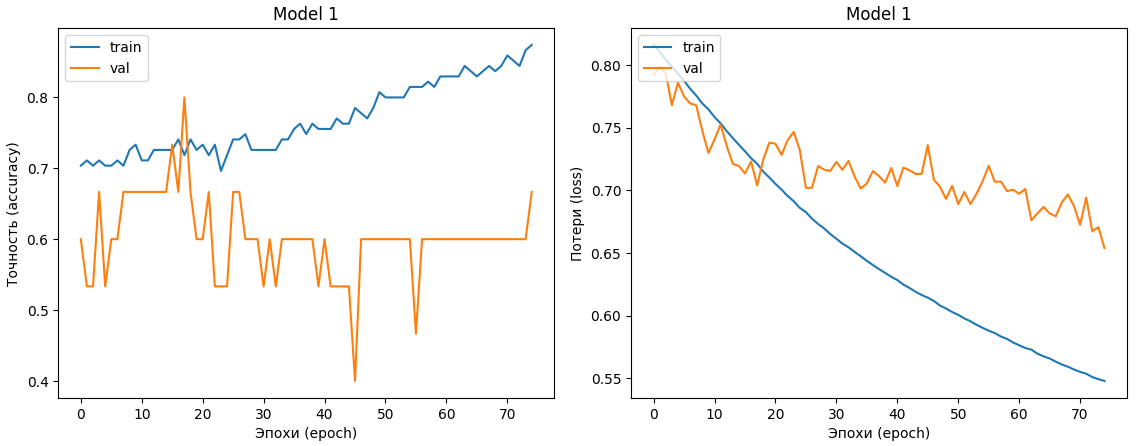


Рисунок 10 – Результаты тестирования модель 1

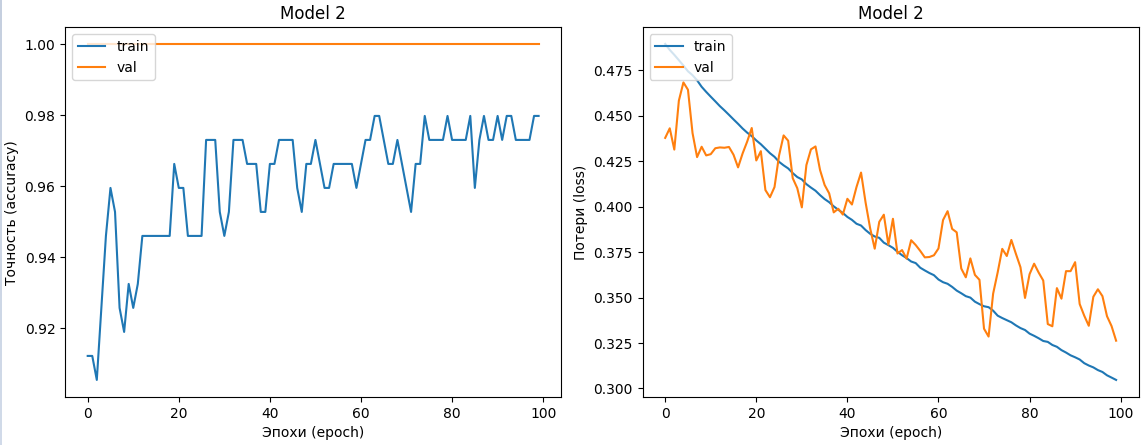


Рисунок 11 – Результаты тестирования модель 2

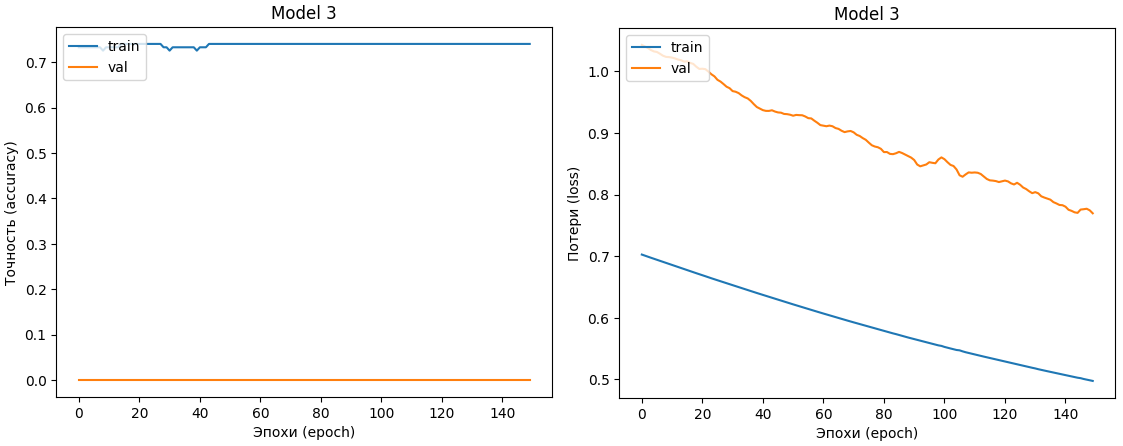


Рисунок 12 – Результаты тестирования модель 3

# **5 Вывод**

В данной лабораторной работе реализовали классификацию сортов растения ирис (Iris Setosa - 0, Iris Versicolour - 1, Iris Virginica - 2) по четырем признакам: размерам пестиков и тычинок его цветков. В ходе обучения моделей были получены графические результаты по которым можно сделать вывод, что вторая модель показывает лучшие результаты, наименьшее количество потерь и наивысшую точность.