

МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра МО ЭВМ

ОТЧЕТ
по лабораторной работе №1
по дисциплине «Искусственные нейронные сети»
Тема: Многоклассовая классификация цветов

Студент гр. 8383

Колмыков В.Д.

Преподаватель

Санкт-Петербург

2021

Цель работы.

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

Ход работы.

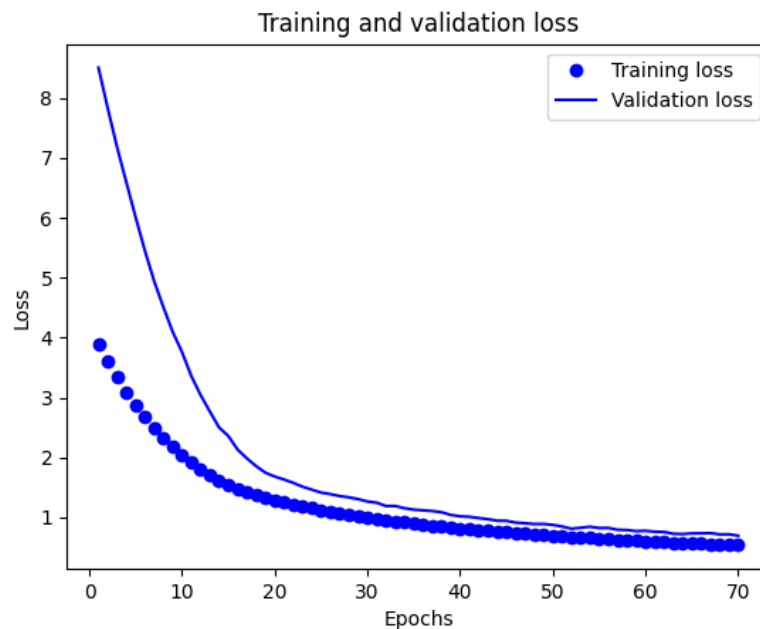
Была написана программа создания и обучения нейросети из методических указаний:

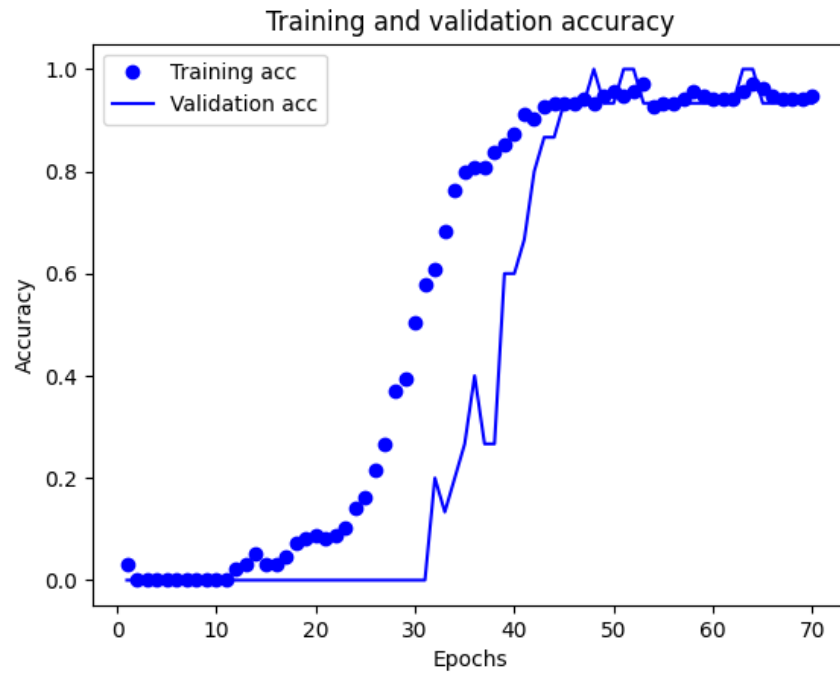
```
model = Sequential()
model.add(Dense(8, activation='relu', input_shape=(4,)))
model.add(Dense(8, activation='relu'))
model.add(Dense(3, activation='softmax'))

model.compile(optimizer='adam', loss='categorical_crossentropy',
metrics=['accuracy'])

history = model.fit(X, dummy_y, epochs=70, batch_size=5,
validation_split=0.2, verbose=False)
```

В результате были получены следующие графики ошибок и точности:



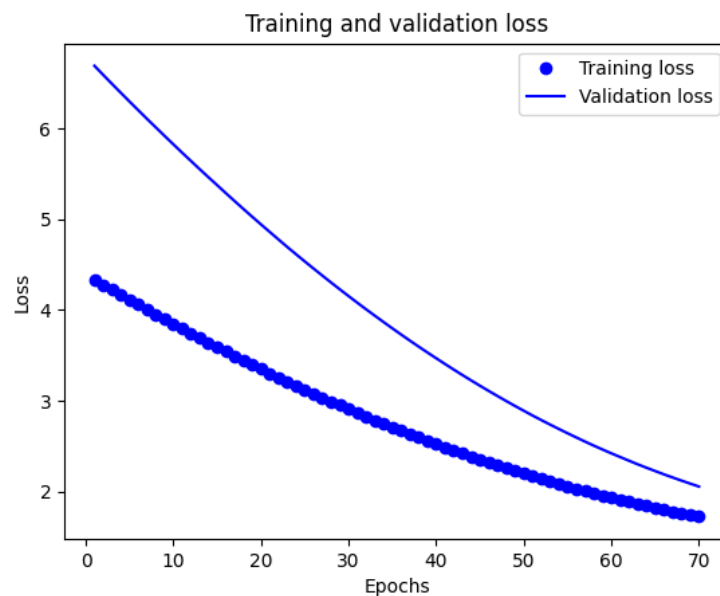


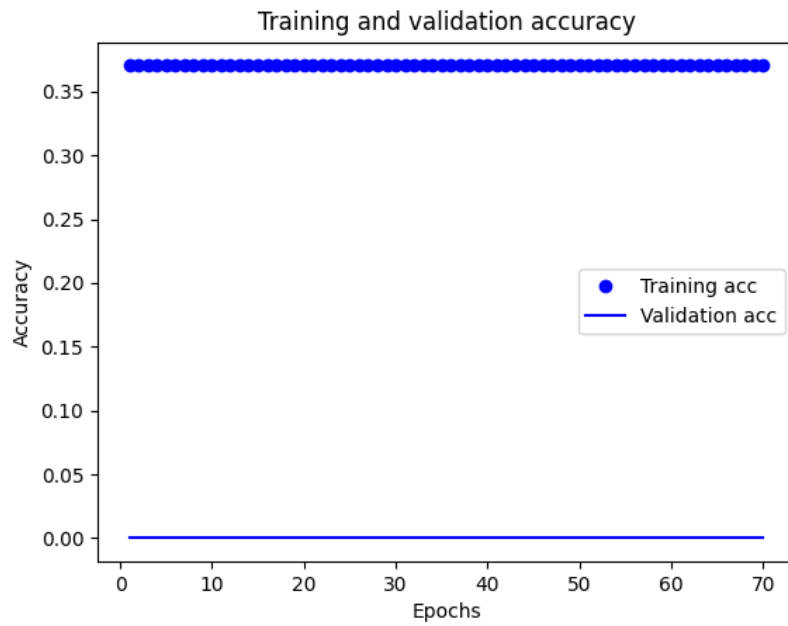
1. Изучение обучение при различных параметрах обучения (параметры функции fit).

- **batch_size**

Параметр определяет размер «батча», т.е. кол-во строк данных, после обработки которых веса нейросети будут корректироваться.

Увеличение приводит к ускорению работы нейросети, так как происходит меньше вычислений, но точность и размер ошибок могут возрасти. Пример работы нейросети с размером «батча» 50:

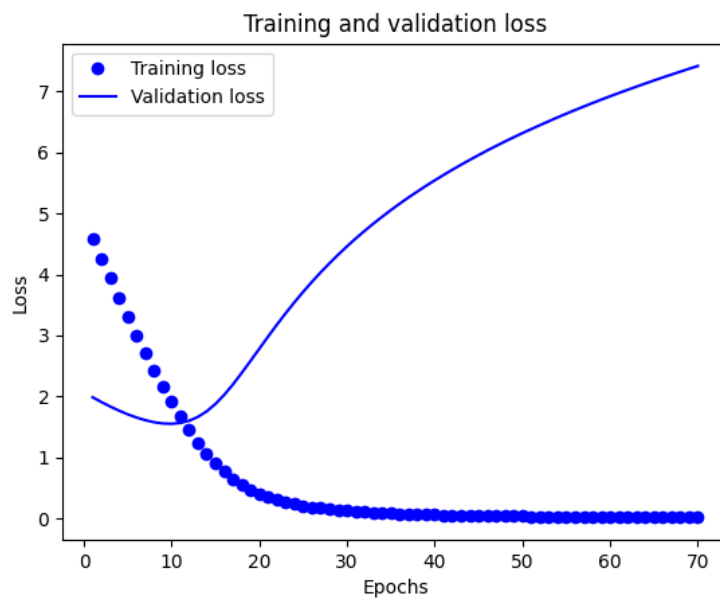


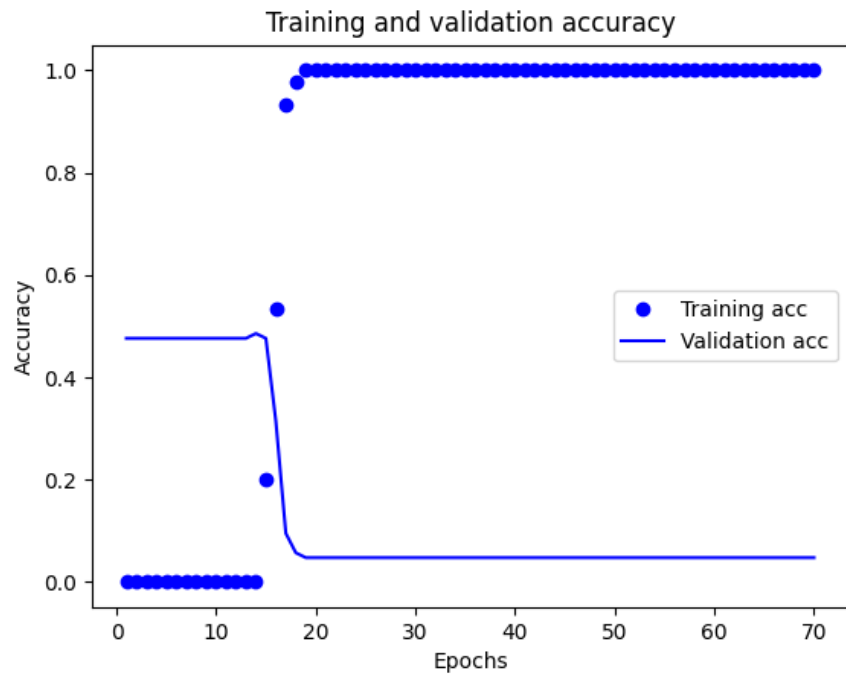


В связи с небольшим размером данных для обучения для обучаемой нейросети размер «батча» был установлен в 5 вместо 10.

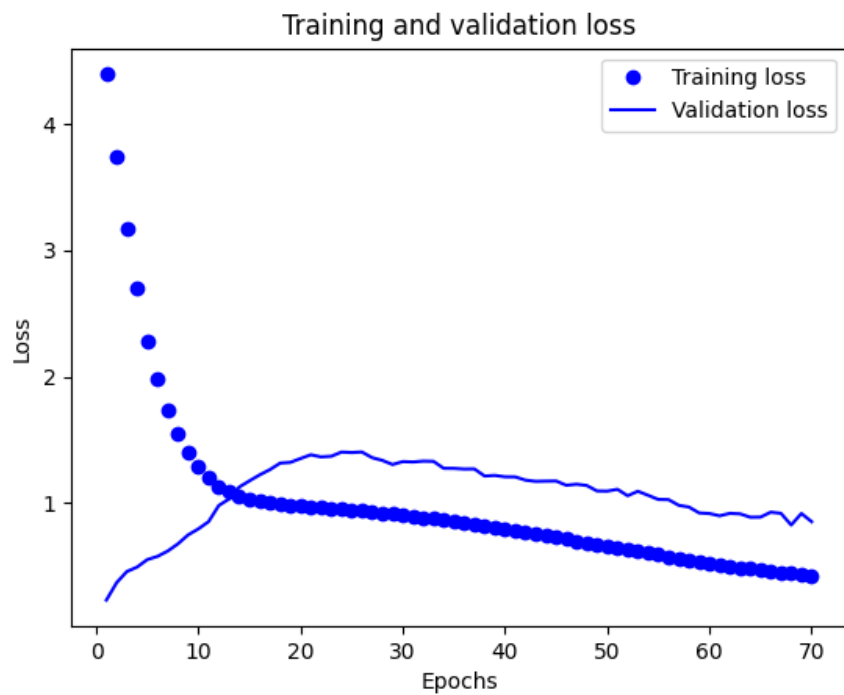
- **validation_split**

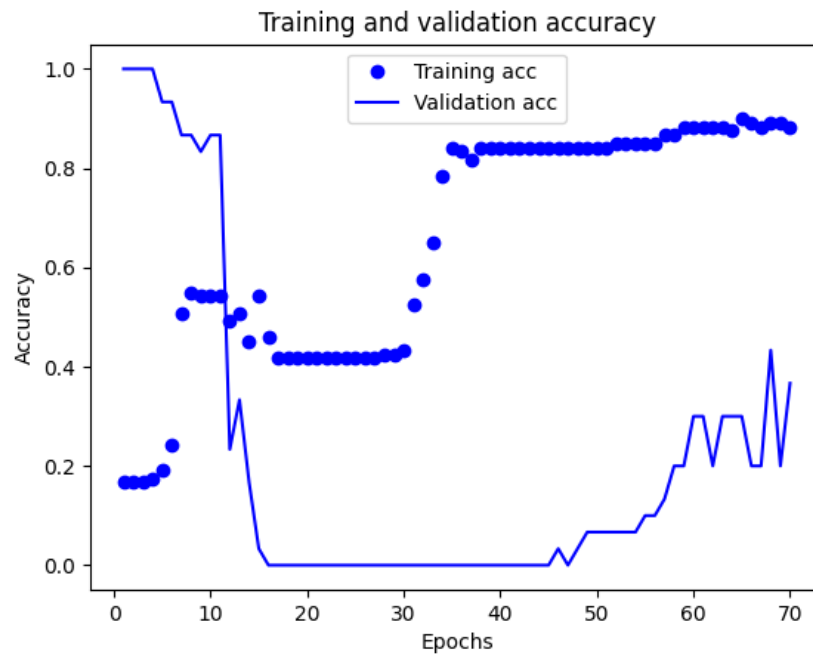
Параметр определяет процент данных, которые будут отведены под валидацию. При увеличении параметра до 0.7 графики не сходятся:





Это логично, так как валидационных данных больше, чем тренировочных. Валидационные данные не должны превышать половины. 0.1 для работы нейросети может оказаться недостаточно. Так как размер заданной выборки данных сравнительно небольшой, он будет установлен в 0.2:





Логично, что ошибки возрасли, а точность уменьшилась. Решать проблему необходимо изменением архитектуры нейросети.

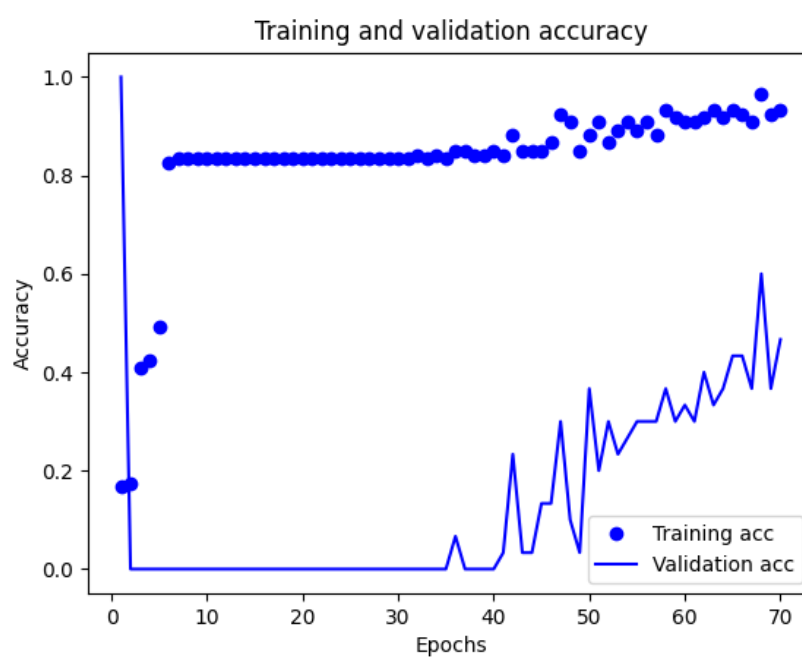
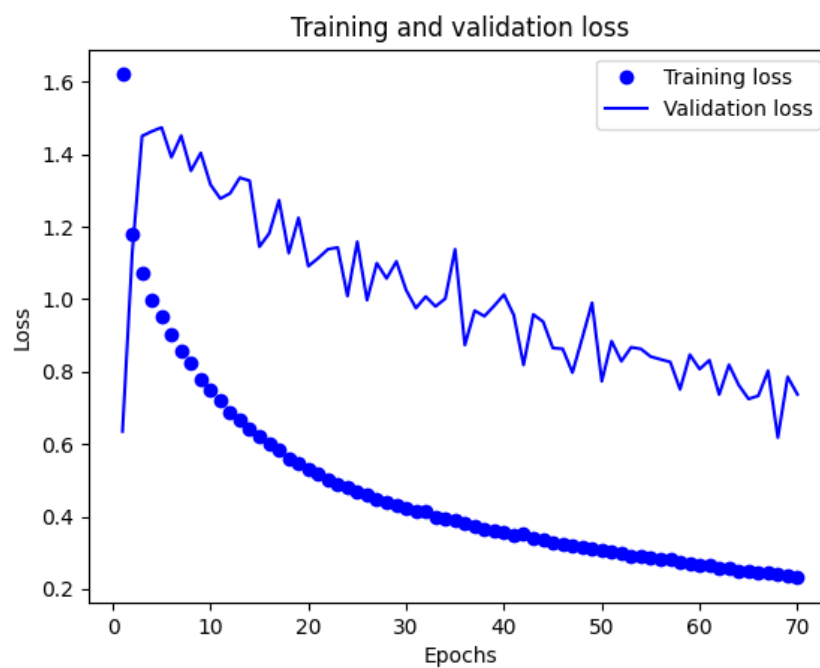
- **epochs**

Параметр определяет кол-во эпох обучения. В некоторых случаях, при слишком большом кол-ве эпох графики могут начать расходиться, т.е. может произойти переобучение. На практике в данной модели получить это даже при размере параметра 1000 не удалось.

2. Изучение архитектуры ИНС.

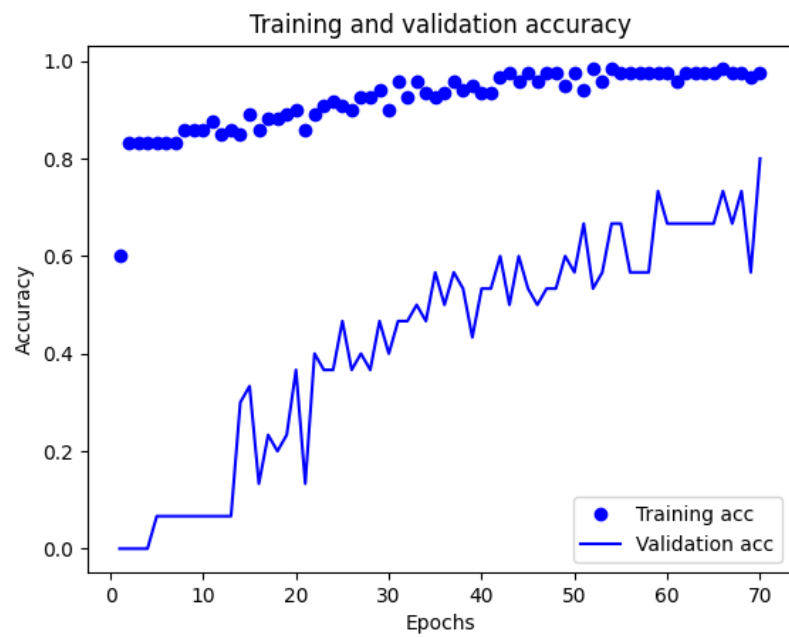
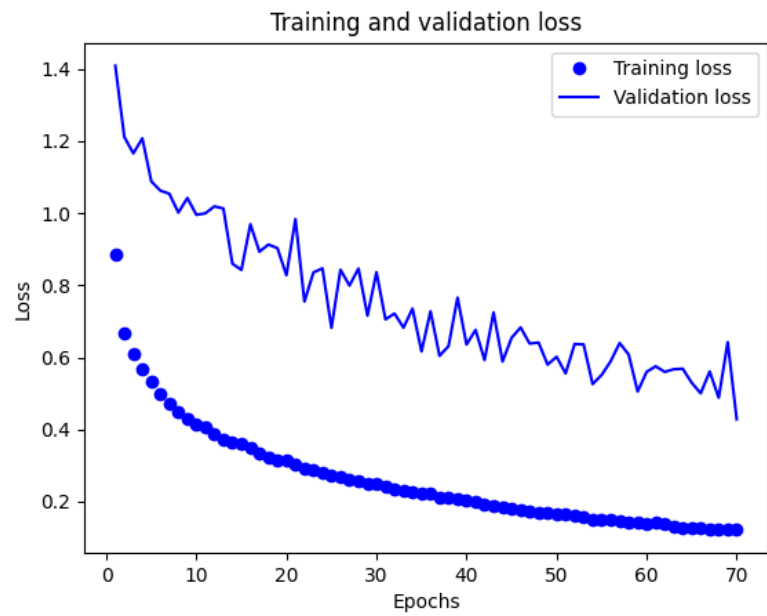
- **Увеличение кол-ва нейронов в слое**

При увеличении кол-ва нейронов в слое до 8:

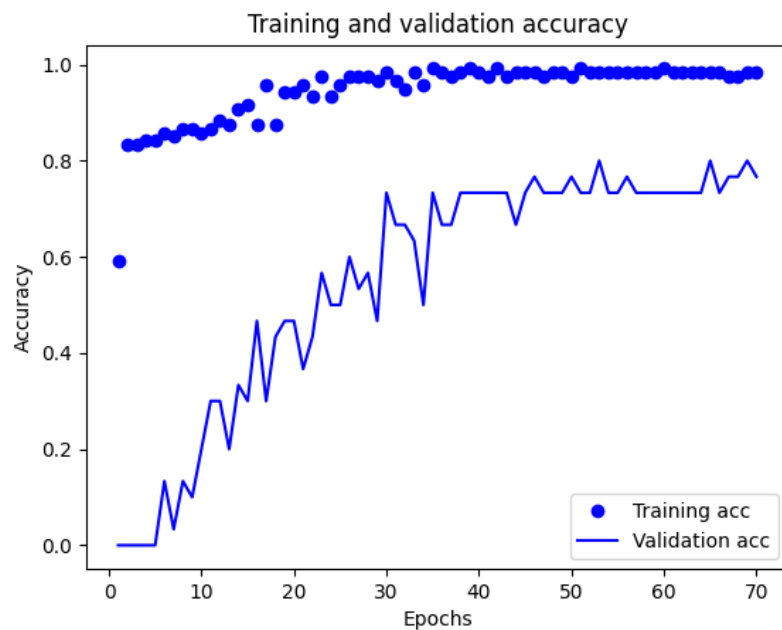
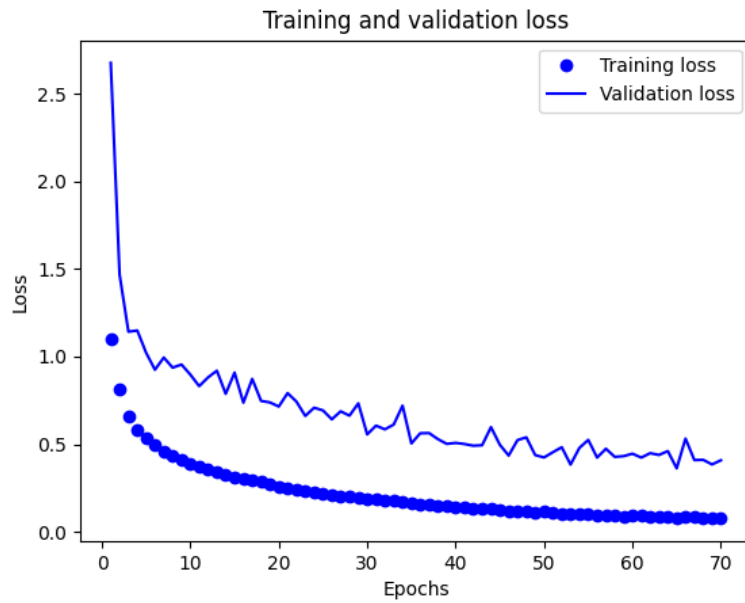


Ошибки уменьшились, точность не сильно изменилась.

Кол-во нейронов было увеличено до 16:



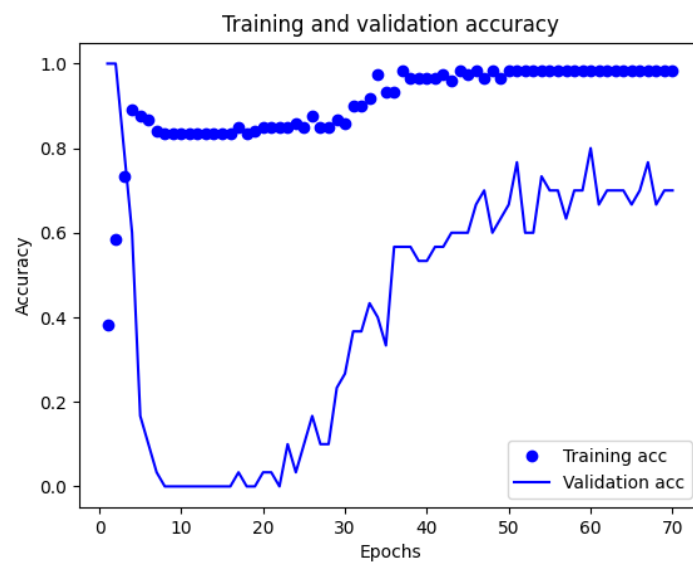
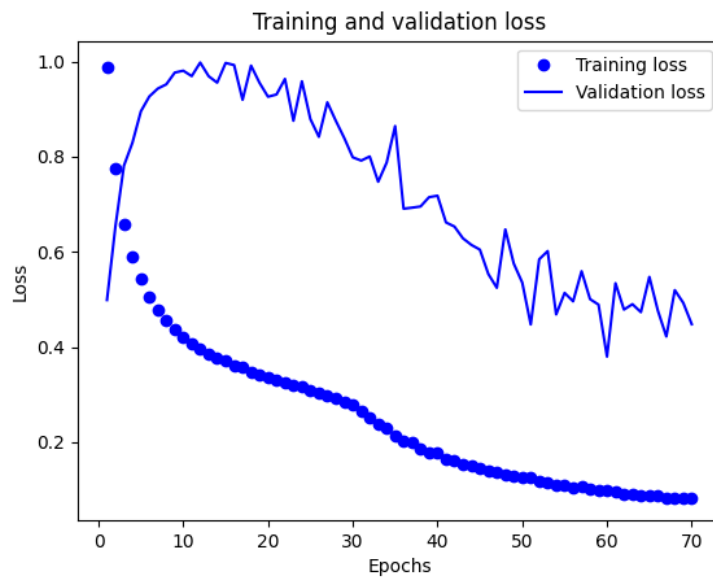
Кол-во нейронов было увеличено до 32:



Видимых изменений в лучшую сторону нет, следовательно можно перестать увеличивать кол-во нейронов.

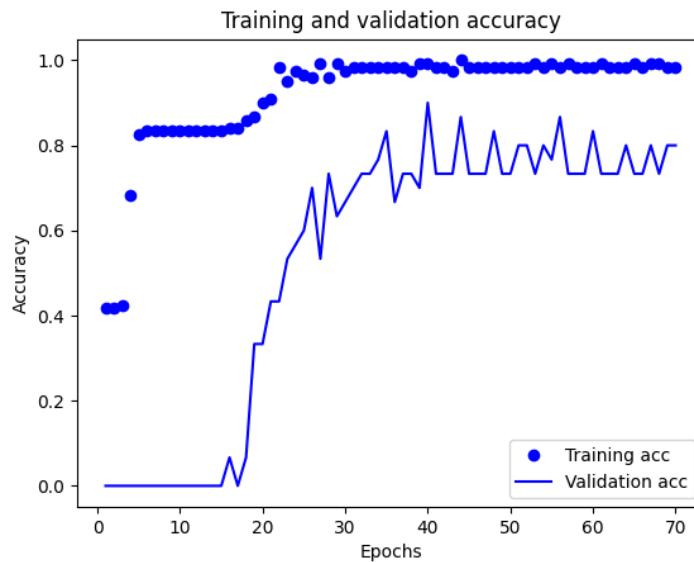
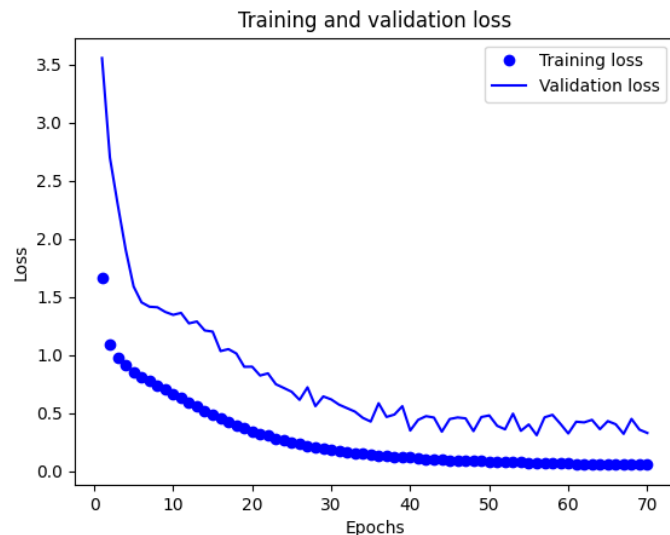
- **Добавление слоя нейронов**

Были взяты два слоя по 4 нейрона:



Видно, что точность и ошибки соизмеримы уровню одного слоя с 32 нейронами.

Были взяты два слоя с 8 нейронами:



Видимых отличий от предыдущей архитектуры не наблюдается.

В итоге можно сделать вывод что лучшая конфигурация нейросети — следующая:

```
model = Sequential()
model.add(Dense(8, activation='relu', input_shape=(4,)))
model.add(Dense(8, activation='relu'))
model.add(Dense(3, activation='softmax'))

model.compile(optimizer='adam', loss='categorical_crossentropy',
metrics=['accuracy'])

history = model.fit(X, dummy_y, epochs=70, batch_size=5,
validation_split=0.2, verbose=False)
```

В качестве аналога можно было бы взять нейросеть с одним слоем из 32 нейронов.

Выводы.

В данной работе была написана нейросеть для классификации сортов ириса по размерам лепестков. Были изучены различные архитектуры ИНС, а также параметры для их обучения.