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

ОТЧЕТ

по лабораторной работе №2

по дисциплине «Искусственные нейронные сети»

Тема: Бинарная классификация отражённых сигналов

Студент гр. 8382 _____ Синельников М.Р.

Преподаватель _____ Жангиров Т.Р.

Санкт-Петербург

2021

Цель работы.

Реализовать классификацию между камнями (R) и металлическими цилиндрами (M) на основе данных об отражении сигналов радара от поверхностей.

60 входных значений показывают силу отражаемого сигнала под определенным углом. Входные данные нормализованы и находятся в промежутке от 0 до 1.

Задачи.

- Ознакомиться с задачей бинарной классификации
- Загрузить данные
- Создать модель ИНС в tf.Keras
- Настроить параметры обучения
- Обучить и оценить модель
- Изменить модель и провести сравнение. Объяснить результаты

Выполнение работы.

Импортируем необходимые для работы классы и функции. Кроме Keras понадобится Pandas для загрузки данных и scikit-learn для подготовки данных и оценки модели.

Набор данных загружается напрямую с помощью pandas. Затем необходимо разделить атрибуты (столбцы) на 60 входных параметров (X) и 1 выходной (Y).

Выходные параметры представлены строками (“R” и “M”), которые необходимо перевести в целочисленные значения 0 и 1 соответственно. Для этого применяется LabelEncoder из scikit-learn.

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; Для задач бинарной классификации применяется функция binary crossentropy.

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 элементов).

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.

Для начала обучим модель с одним скрытым слоем по 60 нейронов и активацией relu.

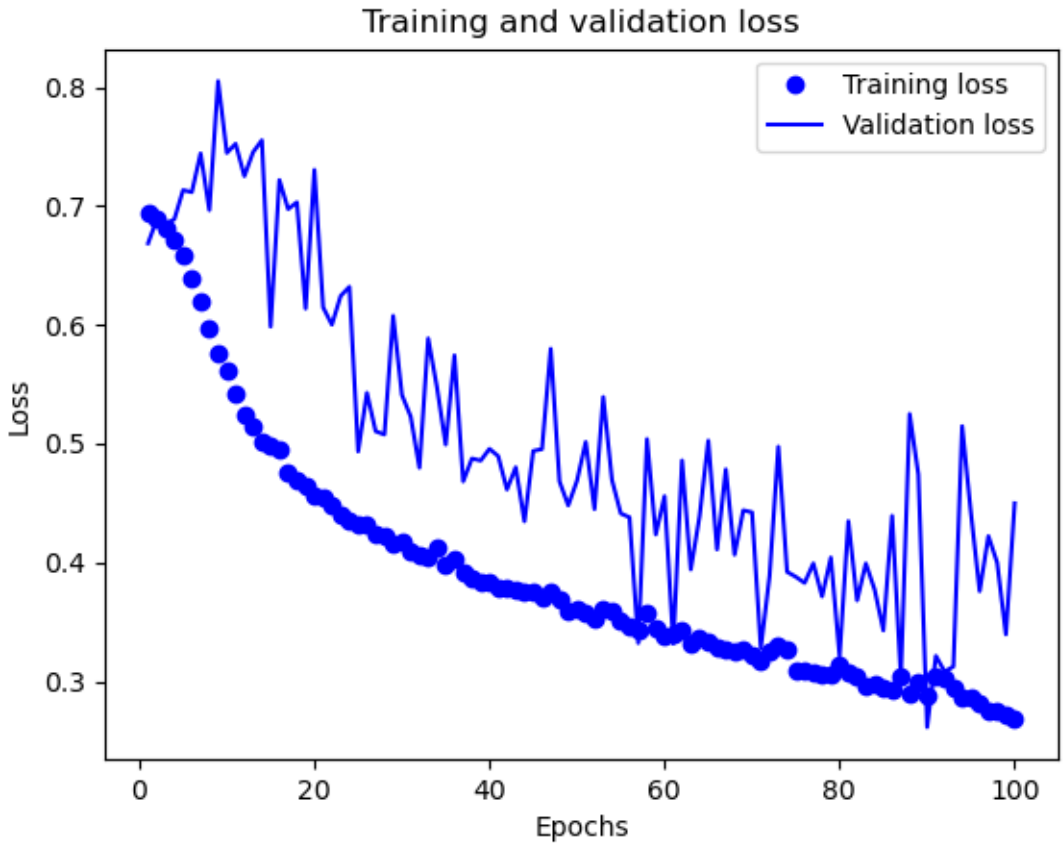

Рисунок 1 – график функции потерь 1 ой модели

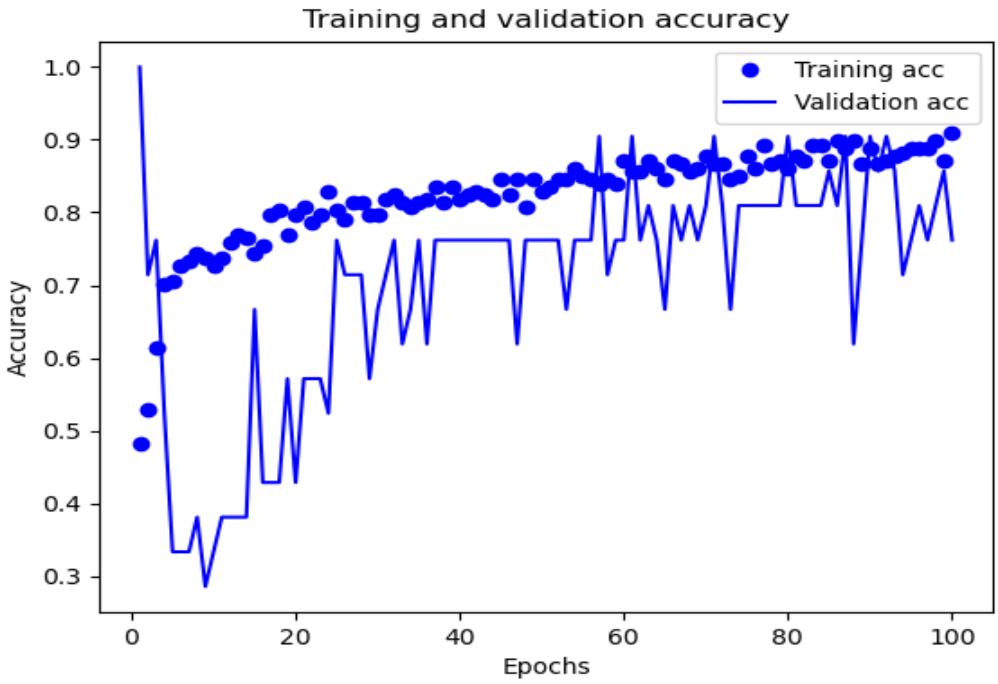

Рисунок 2 — график метрики точности 1 ой модели

Увеличим количество нейронов до 100

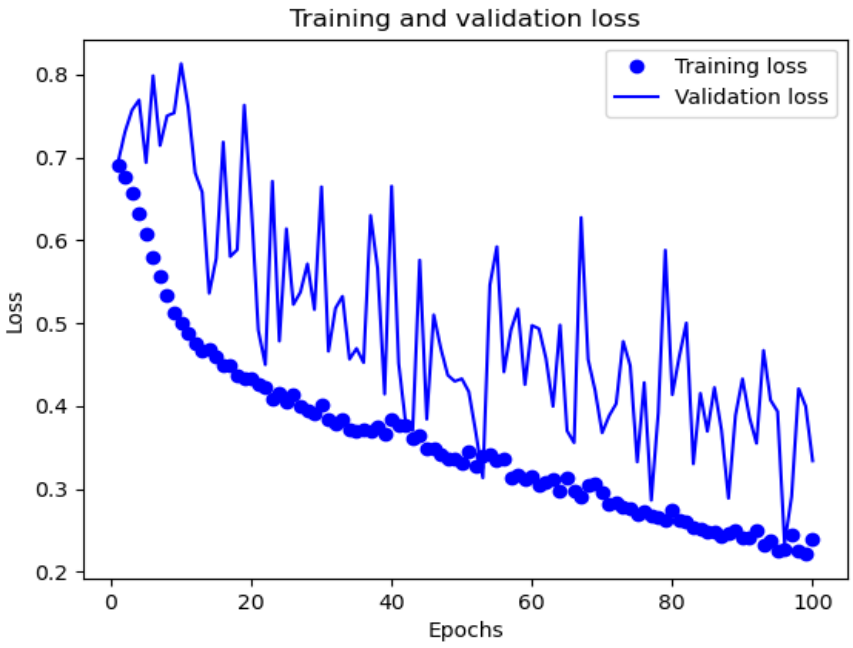

Рисунок 3 — график ошибки 2 ой модели

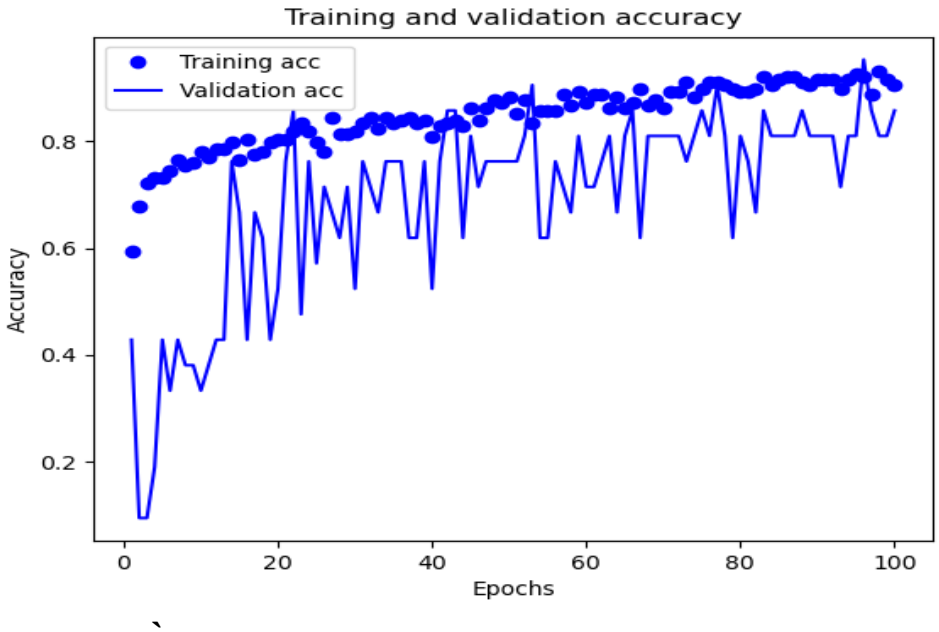

Рисунок 4 — график точности 2 ой модели

Существенных изменений не наблюдается. Добавим ещё один скрытый слой и определим по 60 нейронов на первом и 15.

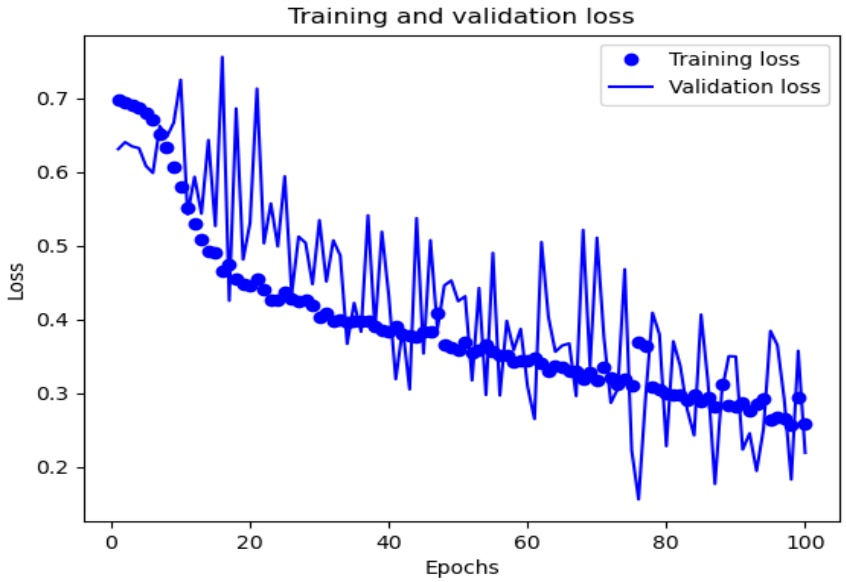

Рисунок 5 — график ошибки 3 ей модели

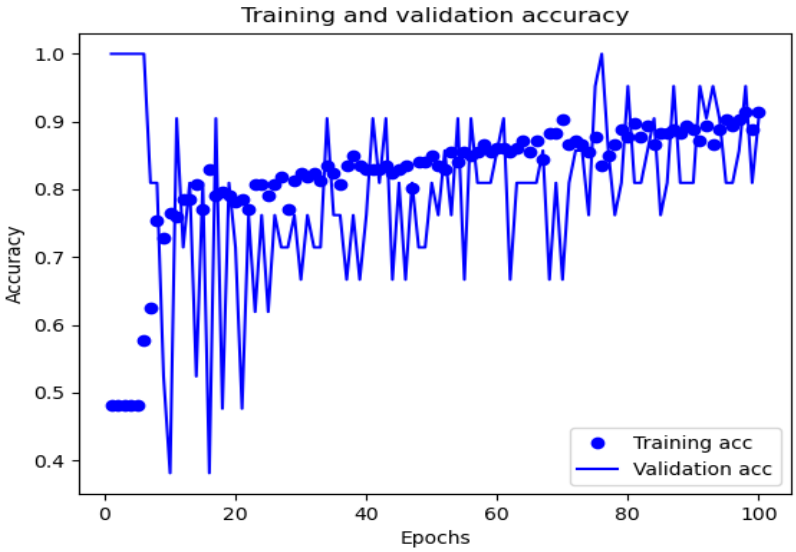

Рисунок 6 — график точности 3 ей модели

Как видно из графиков, значение метрики точность повысилось на валидационном множестве.

Уменьшим входной слой в 2 раза, взяв только первые 30 входные признаки и обучим ту же модель, что и на предыдущем шаге.

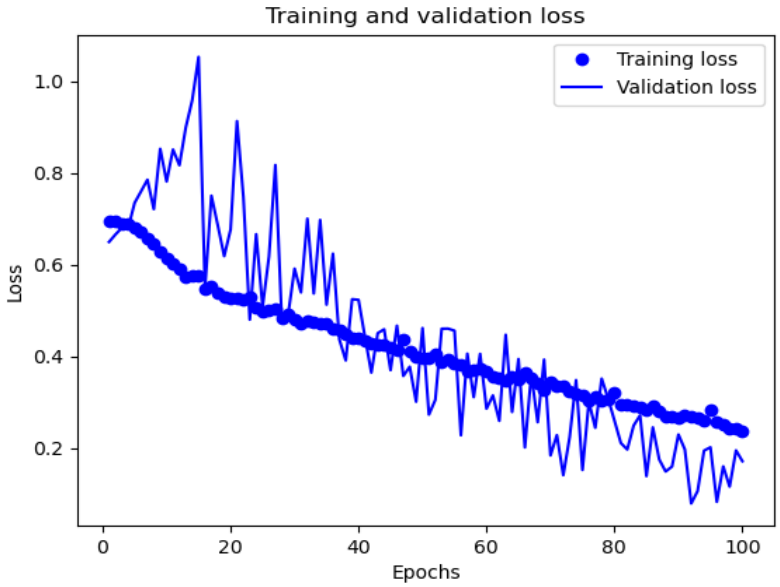

Рисунок 7 — график ошибки 4 ой модели

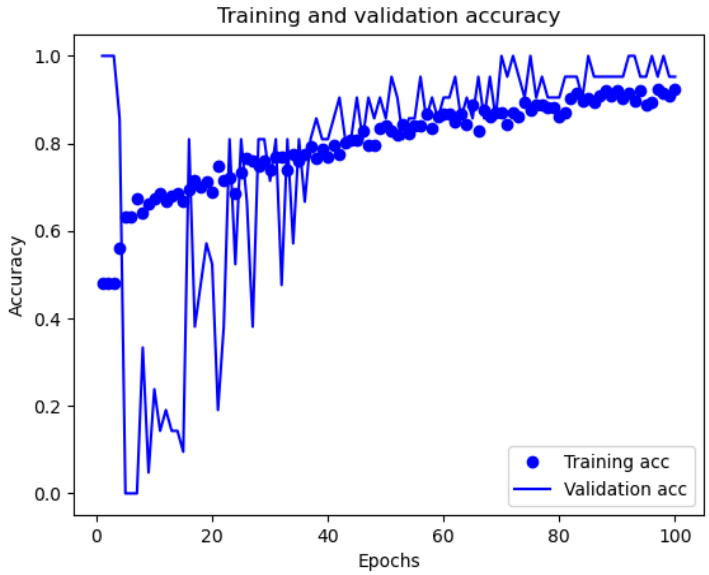

Рисунок 8 — график точности 4 ой модели

Точность модели на валидационном множестве повысилась, так как изначальные входные признаки в некой степени дублировали друг друга и были излишни.

Попробуем выбрать не 30 первых признаков, а сократить размерность входного вектора с помощью метода главных компонент, спроецировав данные в евклидово пространство размерности 30.

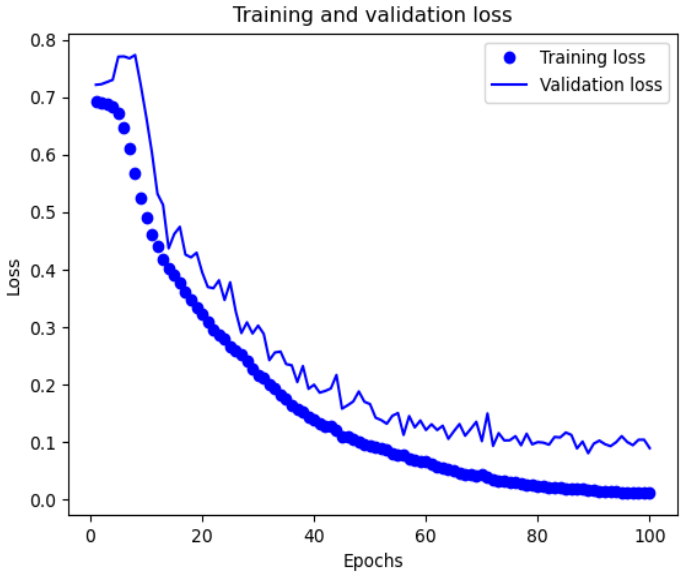

Рисунок 9 — график ошибки 5 ой модели

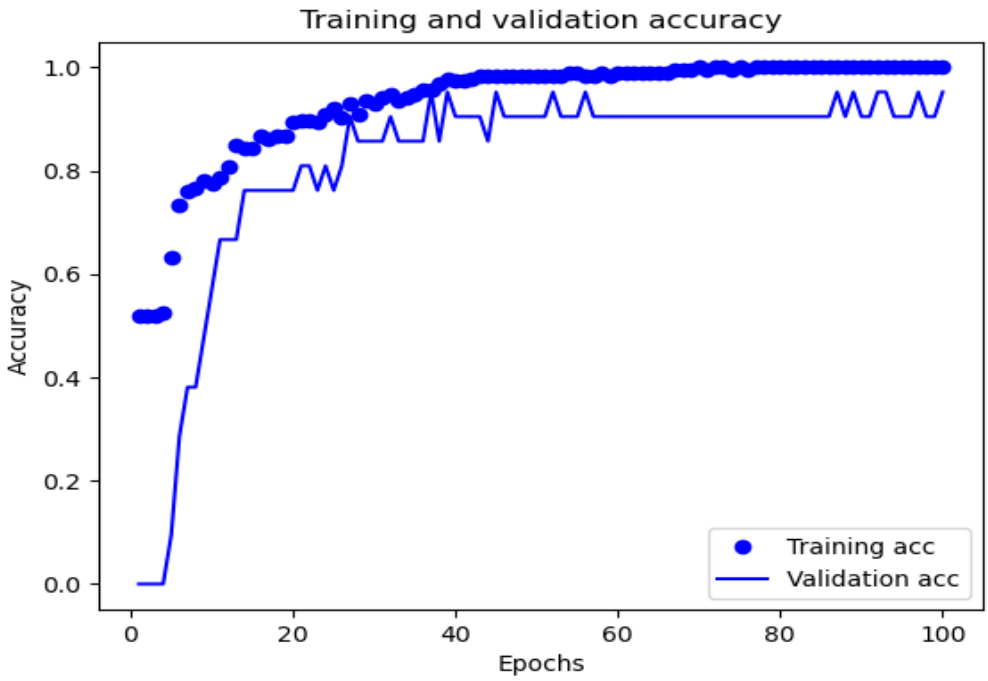

Рисунок 10 — график точности 5 ой модели

Значение метрики почти не изменилось, однако графики обучения выглядят более стабильными.

PS.

Так как обучающих примеров всего 208, то применять нейронную сеть в данном случае может быть не лучшей идеей. Стоит также попробовать более «простой» алгоритм. Но так как структура данных нелинейная, то строить линейную разделяющую поверхность не очень целесообразно. В качестве нелинейного классификатора был выбран метод опорных векторов с rbf ядром. На тестовом множестве метрика точность показывает 0.85.

Выводы.

В результате выполнения лабораторной работы было обучено несколько моделей нс на исходных входных признаках, а также на признаках, сокращённых в два раза. В работе было рассмотрено два способа сокращения входного вектора. Также были представлены графики ошибки и метрики точности каждой модели.