

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

ОТЧЕТ
по лабораторной работе №2
по дисциплине «Искусственные нейронные сети»
Тема: Бинарная классификация отраженных сигналов радара

Студентка гр. 7303

Шишкин И.В.

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.

Был скачан файл с набором данных, на которых нейронная сеть будет обучаться. Затем, была создана модель, состоящая из двух слоев (листинг 1) – она является моделью №1.

Листинг 1 – Модель №1

```
model = Sequential()
model.add(Dense(60, input_dim=60, activation='relu'))
model.add(Dense(1, activation='sigmoid'))
model.compile(optimizer='adam', loss='binary_crossentropy',
metrics=['accuracy'])
H = model.fit(X, encoded_Y, epochs=100, batch_size=10,
validation_split=0.1)
```

Графики этой модели представлены на рис. 1.

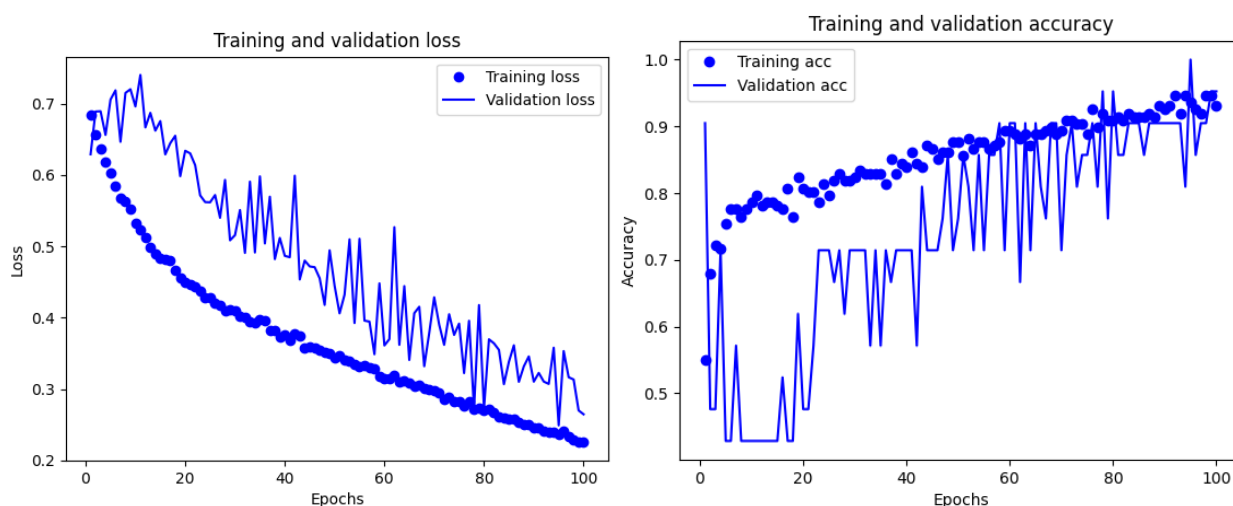


Рисунок 1 – Графики модели №1

От запуска к запуску результаты достаточно сильно меняются. Точность на тестовых данных колебалась от 91 до 96%, на проверочных – от 62 до 100%.

Для модели №2 размер входного слоя был уменьшен в два раза (листинг 2).

Листинг 2 – Модель №2

```
model = Sequential()
model.add(Dense(60, input_dim=30, activation='relu'))
model.add(Dense(1, activation='sigmoid'))
model.compile(optimizer='adam', loss='binary_crossentropy',
metrics=['accuracy'])
H = model.fit(X, encoded_Y, epochs=100, batch_size=10,
validation_split=0.1)
```

Графики этой модели представлены на рис. 2.

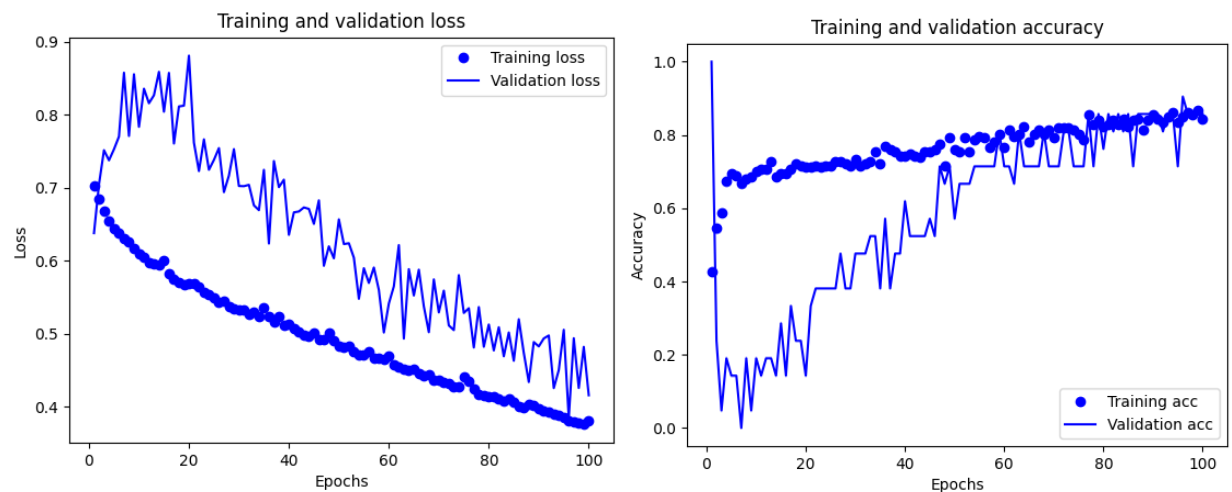


Рисунок 2 – Графики модели №2

Результат почти такой же, но точность на тестовых в среднем меньше, чем до этого. Точность на тестовых данных колеблется от 81 до 88%, на проверочных – от 76 до 95%.

Для модели №3 был добавлен промежуточный (скрытый) слой Dense в архитектуру сети с 15 нейронами (листинг 3).

Листинг 3 – Модель №3

```
model = Sequential()
model.add(Dense(60, input_dim=30, activation='relu'))
model.add(Dense(15, activation='relu'))
model.add(Dense(1, activation='sigmoid'))
model.compile(optimizer='adam', loss='binary_crossentropy',
metrics=['accuracy'])
H = model.fit(X, encoded_Y, epochs=100, batch_size=10,
validation_split=0.1)
```

Графики этой модели представлены на рис. 3.

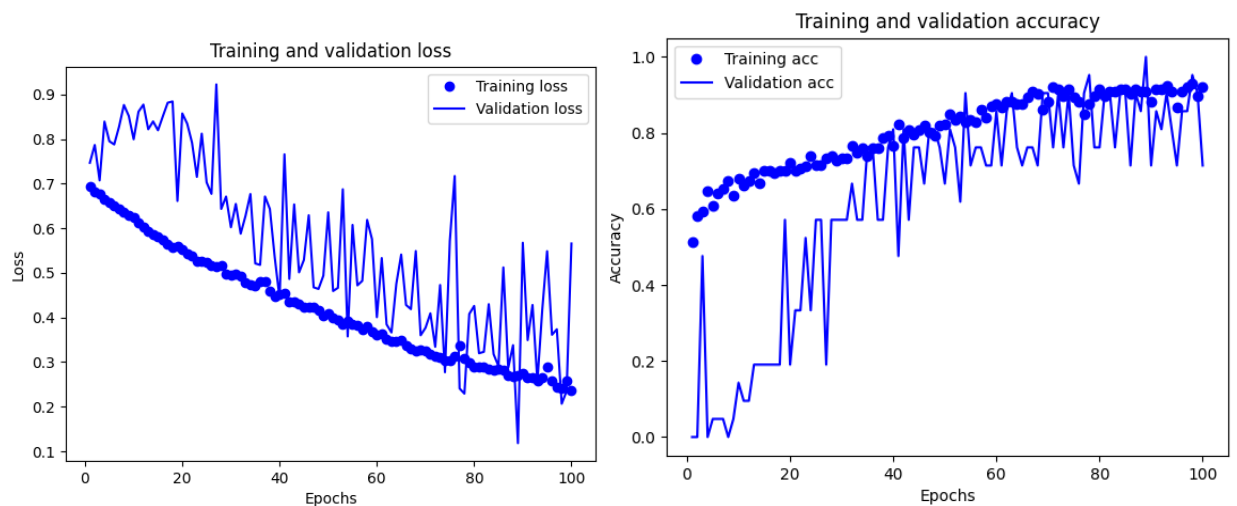


Рисунок 3 – Графики модели №3

Потерь уже намного меньше. Точность на тестовых данных колеблется от 94 до 99%, на проверочных – от 86 до 100%.

Для модели №4 уменьшим число нейронов на первом скрытом слое в 2 раза (листинг 4).

Листинг 4 – Модель №4

```
model = Sequential()
model.add(Dense(30, input_dim=30, activation='relu'))
model.add(Dense(15, activation='relu'))
model.add(Dense(1, activation='sigmoid'))
model.compile(optimizer='adam', loss='binary_crossentropy',
metrics=['accuracy'])
H = model.fit(X, encoded_Y, epochs=100, batch_size=10,
validation_split=0.1)
```

Графики этой модели представлены на рис. 4.

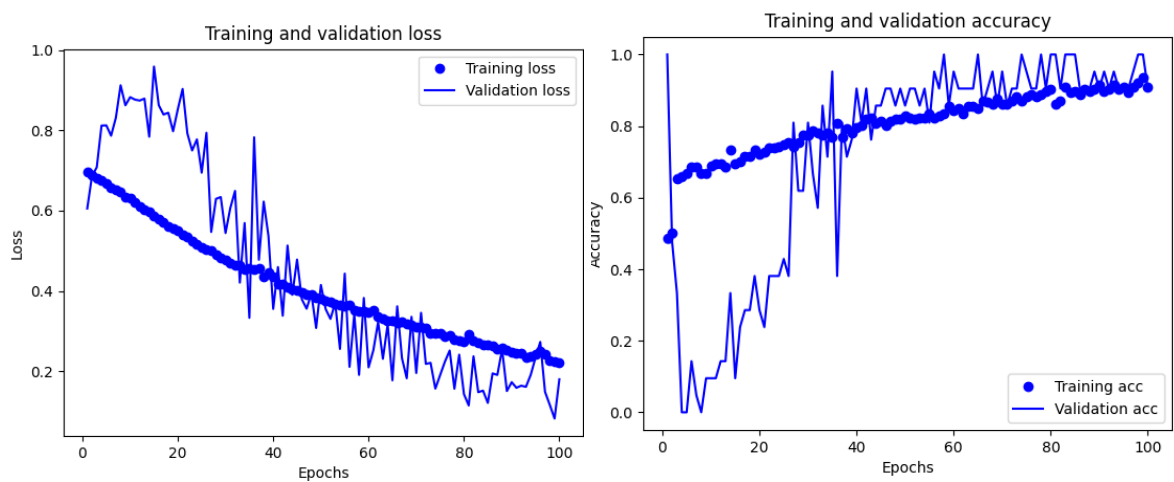


Рисунок 4 – Графики модели №4

Потери и точность хуже, чем у предыдущей модели.

Для модели №5 вернем количество нейронов на первом слое до 60, и добавим еще один скрытый слой (листинг 5).

Листинг 5 – Модель №5

```
model = Sequential()
model.add(Dense(60, input_dim=30, activation='relu'))
model.add(Dense(15, activation='relu'))
model.add(Dense(5, activation='relu'))
model.add(Dense(1, activation='sigmoid'))
model.compile(optimizer='adam', loss='binary_crossentropy',
metrics=['accuracy'])
H = model.fit(X, encoded_Y, epochs=100, batch_size=10,
validation_split=0.1)
```

Графики этой модели представлены на рис. 5.

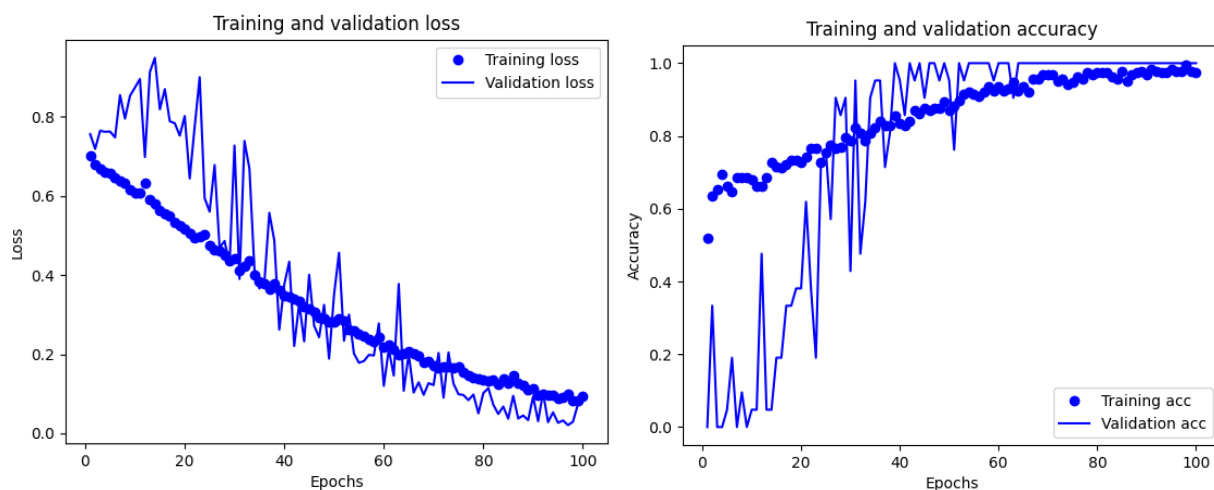


Рисунок 5 – Графики модели №5

В данном случае потери становятся еще меньше (на обучающих данных в среднем 0.05), точность еще выше, причем точность на проверочных данных часто достигает 100% (в 16 запусках из 20).

Выводы.

Изучено влияние кол-ва нейронов на слое на результат обучения модели. Изучено влияние кол-ва слоев на результат обучения модели. Построены графики ошибки и точности в ходе обучения. Лучшей моделью оказалась последняя под номером 5, в которой 3 скрытых слоя.