

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

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

Студент гр. 8383

Степанов В.Д.

Преподаватель

Жангиров Т.Р.

Санкт-Петербург

2021

Цель.

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.

Выполнения работы.

Была реализована модель ИНС из методических материалов.

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)
```

Результаты запуска модели:

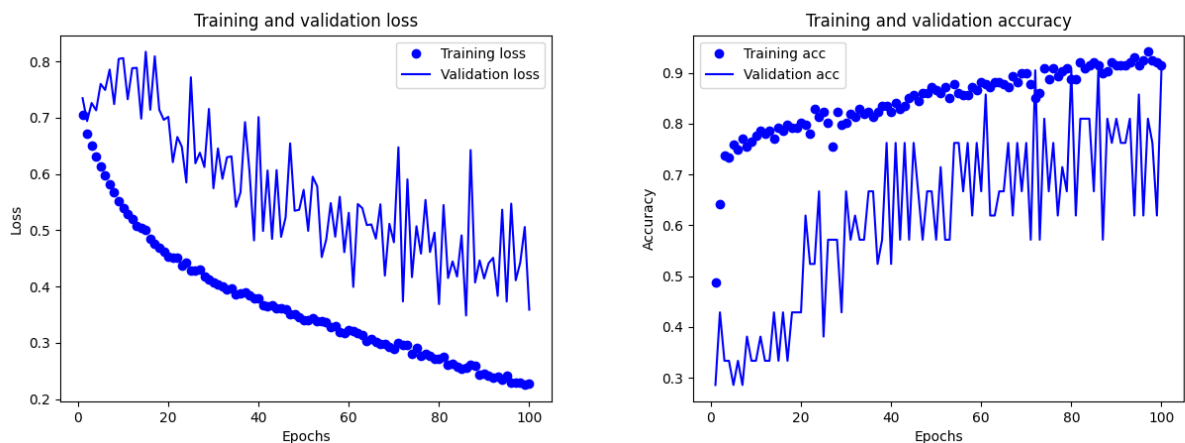


Рисунок 1 – График ошибки и точности модели №1

Таблица 1 – Результаты запусков модели №1

Параметр	Min	Max
loss	0.1656	0.2389
accuracy	0.9125	0.9791
val_loss	0.3081	0.4915
val_accuracy	0.7143	0.9848

В наборе присутствует некоторая избыточность, уменьшим входной набор взяв первые 30 параметров. Так же изменим количество входных нейронов до 30.

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)
```

Результаты запуска модели:

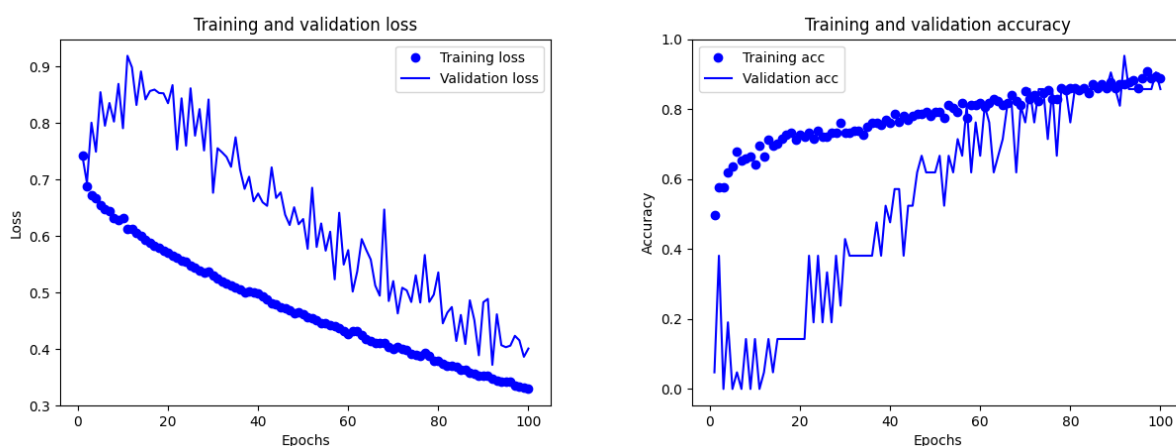


Рисунок 2– График ошибки и точности модели №2

Таблица 2 – Результаты запусков модели №2

Параметр	Min	Max
loss	0.3058	0.3619
accuracy	0.8530	0.8775
val_loss	0.3569	0.4491
val_accuracy	0.8571	0.9048

Сравнивая рис. 1 и 2 и табл. 1 и 2 видно, что амплитуда колебаний ошибки и точности модели №2 меньше модели №1. Так же у модели №2 немного лучше показатель точности на тестовых данных.

Так же модель была обучена и протестирована на наборе, в который входят последние 30 параметров. Точность сети на тестовых данных была сравнима со случайным выбором. На наборе из 30 параметров, взятых из середины, точность сети была меньше, чем случайный выбор. Можно сделать вывод, что для первая половина параметров имеет большую значимость чем другие.

На основе модели №2 и наборе данных, состоящих из первой половины параметров, добавим еще один скрытый слой, состоящий из 15 нейронов.

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)
```

Результаты запуска модели:

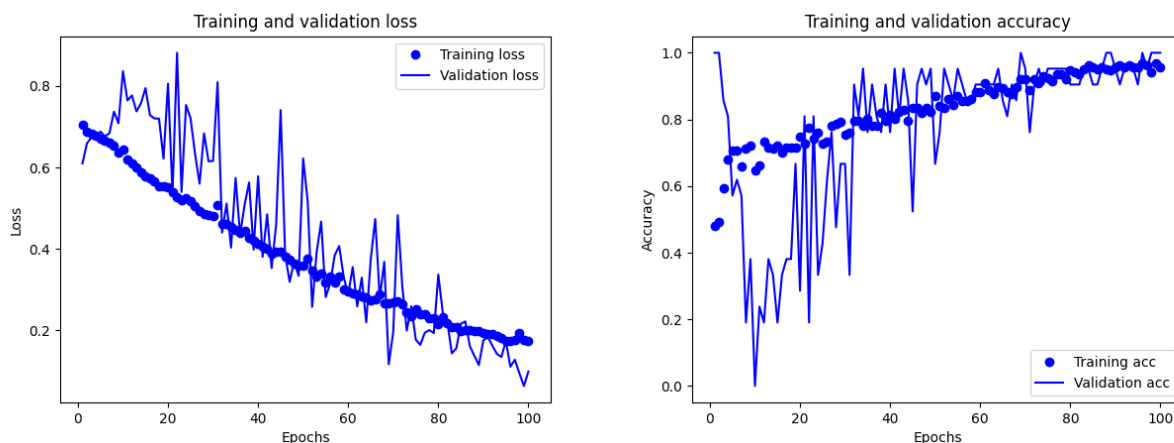


Рисунок 3– График ошибки и точности модели №3

Таблица 3 – Результаты запусков модели №3

Параметр	Min	Max
loss	0.0888	0.1877
accuracy	0.9355	0.9905
val_loss	0.0587	0.2543
val_accuracy	0.9048	1.0000

Сравнивая рис. 1-3 и табл. 1-3 видно, модель № 3 показывает наилучшие результаты по сравнению с предыдущими моделями. Это связано с добавлением еще одного скрытого слоя в модель, что позволило находить закономерности в комбинациях данных.

В модели №3 попробуем уменьшить количество нейронов на первом скрытом слое с 60 до 30.

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)
```

Результаты запуска модели:

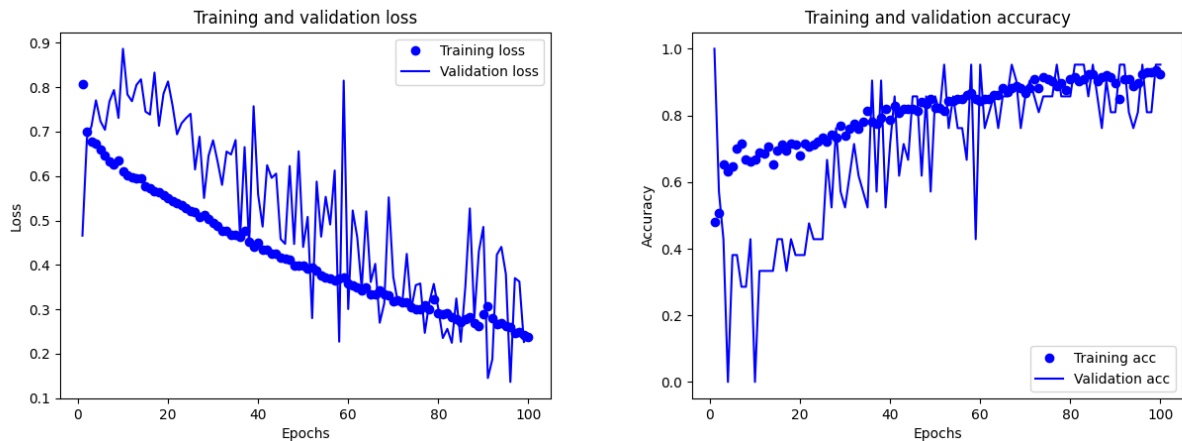


Рисунок 4– График ошибки и точности модели №4

Таблица 4 – Результаты запусков модели №4

Параметр	Min	Max
loss	0.2052	0.3544
accuracy	0.8559	0.9242
val_loss	0.1167	0.3368
val_accuracy	0.8571	1.0000

Сравнивая рис. 3 и 4 и табл. 3 и 4 видно, что модель №4 показала результаты хуже, чем модель №3. Можем сделать вывод, что уменьшение нейронов не даст нам лучшего результата.

На основе модели №3 построим новую, увеличив число нейронов во втором скрытом слое с 15 до 30 нейронов.

Модель №5

```

model = Sequential()
model.add(Dense(60, input_dim=30, activation='relu'))
model.add(Dense(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)

```

Результаты запуска модели:

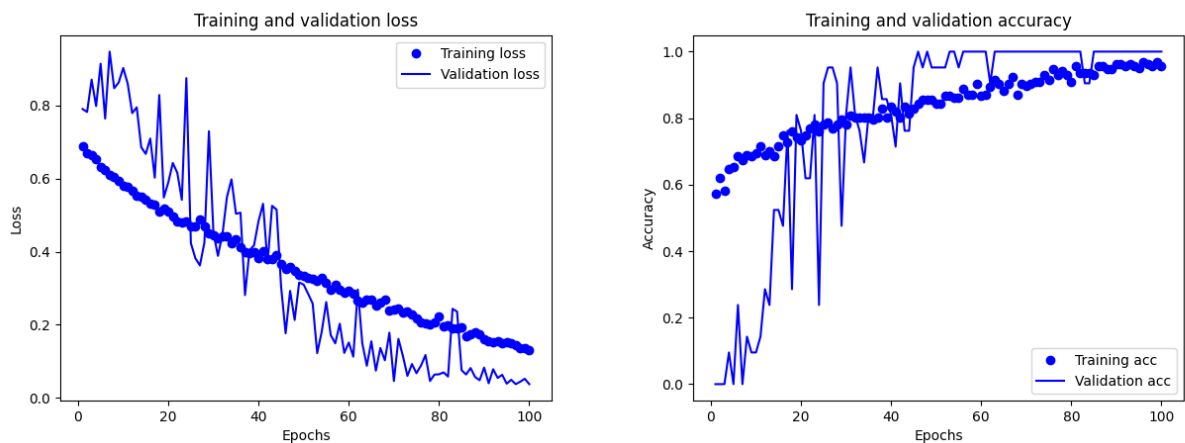


Рисунок 5– График ошибки и точности модели №5

Таблица 5– Результаты запусков модели №5

Параметр	Min	Max
loss	0.0707	0.1130
accuracy	0.9653	0.9895
val_loss	0.0375	0.2096
val_accuracy	0.9524	1.0000

Сравнивая рис. 1-5 и табл. 1-5 видно, что модель №5 показывает лучшие результаты по сравнению с остальными. В нашем случае увеличение количества нейронов, привело к увеличению точности модели. Больше

увеличивать количество нейронов не стоит, так как это может привести к переобучению ИНС.

Выводы.

В ходе лабораторной работы была построена модель классификации между камнями и металлическими цилиндрами на основе данных об отражении сигналов радара от поверхностей. Были изучены влияние количества нейронов и слоев на обучение модели.