

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. 7383

Александров Р.А.

Преподаватель

Жукова Н.А.

Санкт-Петербург

2020

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.

Постановка задачи.

1. Ознакомиться с задачей бинарной классификации
2. Загрузить данные
3. Создать модель ИНС в Keras
4. Настроить параметры обучения
5. Обучить и оценить модель
6. Изменить модель и провести сравнение. Объяснить результаты

Требования.

1. Изучить влияние кол-ва нейронов на слое на результат обучения модели
2. Изучить влияние кол-ва слоев на результат обучения модели
3. Построить графики ошибки и точности в ходе обучения
4. Провести сравнение полученных сетей, объяснить результат

Выполнение работы.

Начнем использовать первый слой с 60 нейронами.

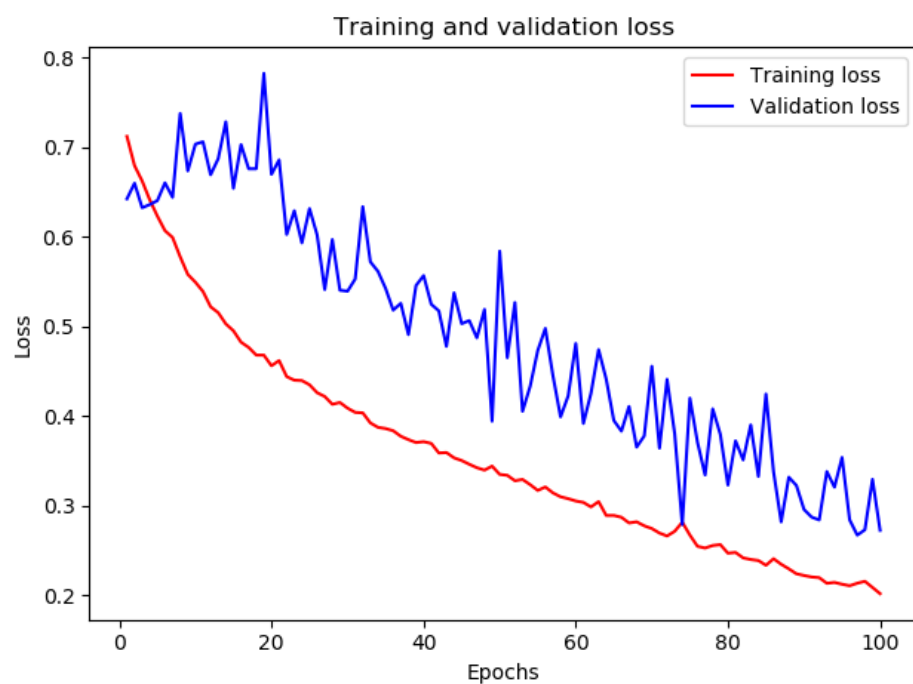


Рисунок 1 – Ошибки с входным слоем на 60 нейронов

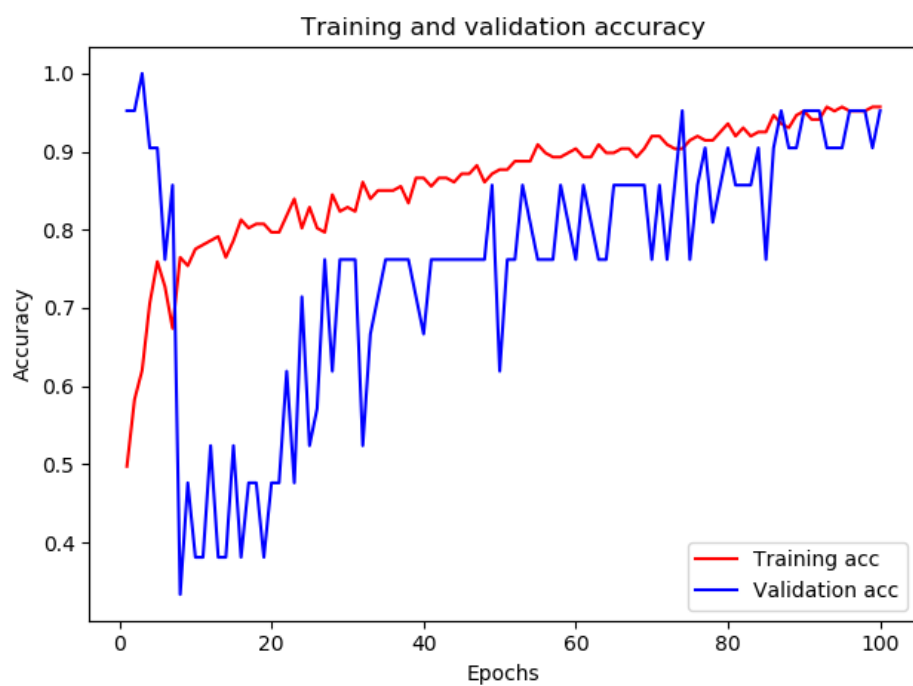


Рисунок 2 – Точность с входным слоем на 60 нейронов

Уменьшим первый слой в 2 раза.

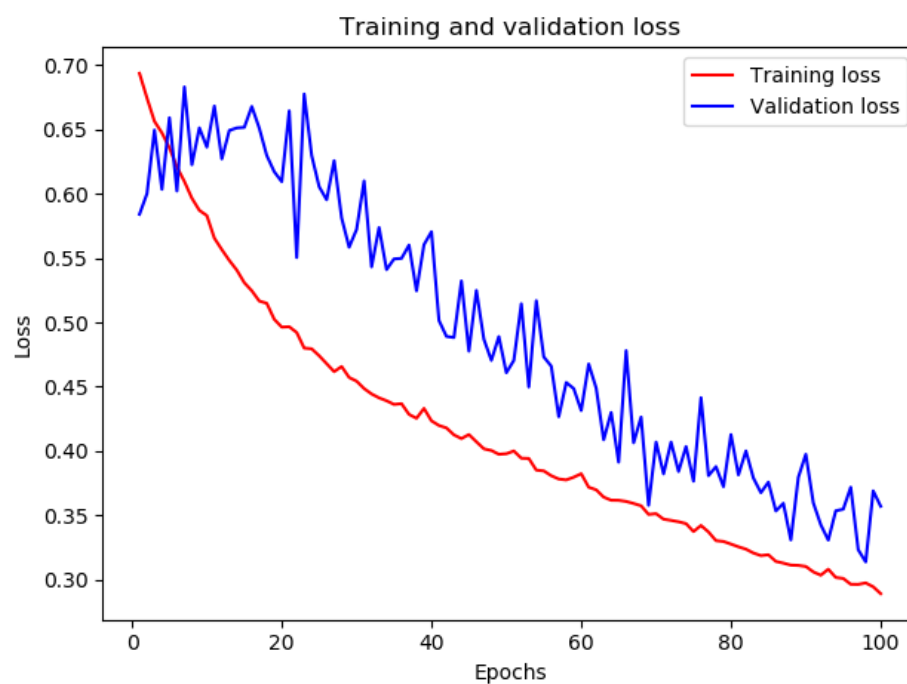


Рисунок 3 – Ошибки с входным слоем на 30 нейронов

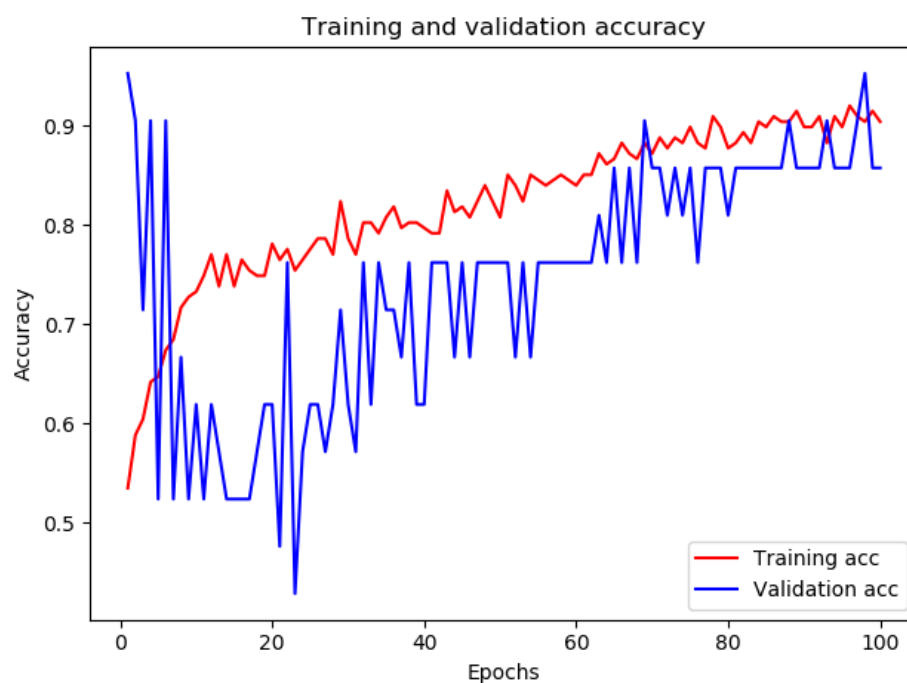


Рисунок 4 – Точность с входным слоем на 30 нейронов

Добавим промежуточный слой на 15 нейронов.

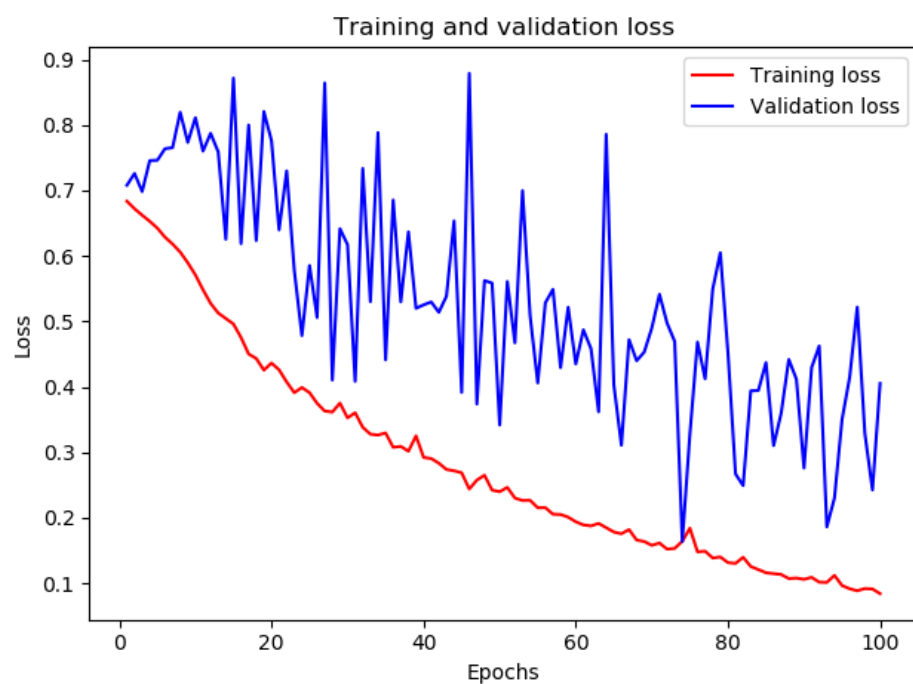


Рисунок 5 – Ошибки с входным слоем на 30 нейронов и промежуточным слоем на 15

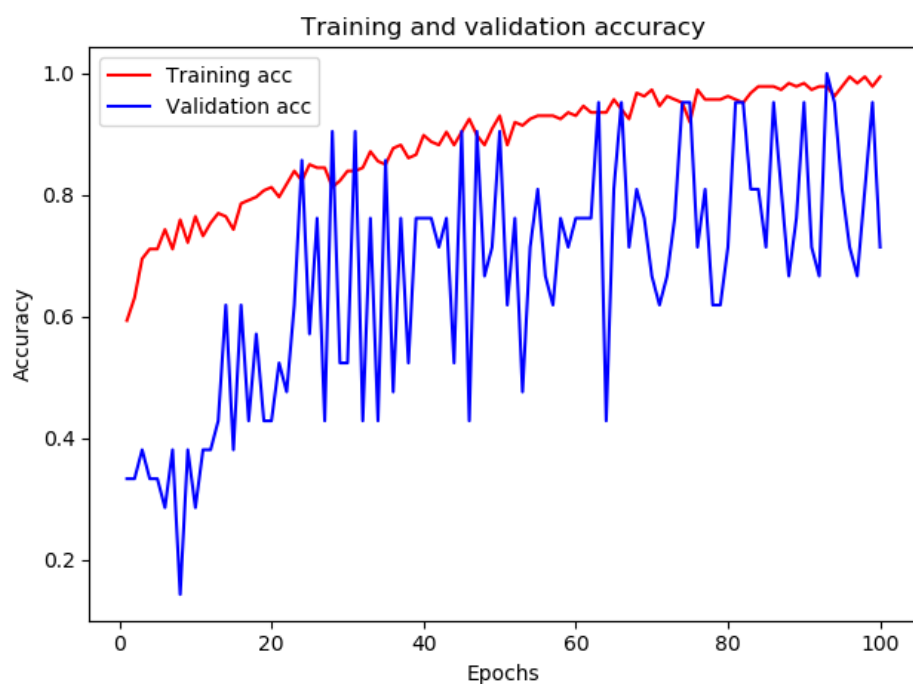


Рисунок 6 – Точность с входным слоем на 30 нейронов и промежуточным слоем на 15

Выводы.

В ходе работы было выявлено, что количество нейронов на входном слое влияет на количество признаков, с которым будет работать сеть. Второй слой нужен для рассмотрения комбинаций результатов работы первого слоя.

ПРИЛОЖЕНИЕ А

ИСХОДНЫЙ КОД ПРОГРАММЫ

```
import pandas
from tensorflow.keras.layers import Dense
from tensorflow.keras.models import Sequential
from sklearn.preprocessing import LabelEncoder
import matplotlib.pyplot as plt

dataframe = pandas.read_csv("sonar.csv", header=None)
dataset = dataframe.values
X = dataset[:, 0:60].astype(float)
Y = dataset[:, 60]

encoder = LabelEncoder()
encoder.fit(Y)
encoded_Y = encoder.transform(Y)

model = Sequential()
model.add(Dense(30, input_dim=60, activation='relu'))
model.add(Dense(15, input_dim = 60, activation='relu'))
model.add(Dense(1, activation='sigmoid'))

model.compile(optimizer='adam',          loss='binary_crossentropy',
metrics=['accuracy'])
history = model.fit(X, encoded_Y, epochs=100, batch_size=10,
validation_split=0.1)

history_dict = history.history
loss_values = history_dict['loss']
val_loss_values = history_dict['val_loss']
acc = history_dict['accuracy']
val_acc = history_dict['val_accuracy']
epochs = range(1, len(loss_values) + 1)

plt.plot(epochs, loss_values, 'r', label='Training loss')
plt.plot(epochs, val_loss_values, 'b', label='Validation loss')
plt.title('Training and validation loss')
plt.xlabel('Epochs')
plt.ylabel('Loss')
plt.legend()
plt.show()

plt.clf()
plt.plot(epochs, acc, 'r', label='Training acc')
plt.plot(epochs, val_acc, 'b', label='Validation acc')
```

```
plt.title('Training and validation accuracy')
plt.xlabel('Epochs')
plt.ylabel('Accuracy')
plt.legend()
plt.show()
```