

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

Студент гр. 8382

Гордиенко А.М.

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.

Порядок выполнения работы.

- 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

Требования к выполнению задания.

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

Ход работы.

Была создана модель. Модель состоит из одного промежуточного слоя с функцией 'relu' из 60 нейронов и выходным слоем с функцией 'sigmoid'.

В качестве параметров обучения были выбраны: Оптимизатор – adam, функция потерь - binary_crossentropy, метрика – точность, количество эпох – 100, размер партии – 10.

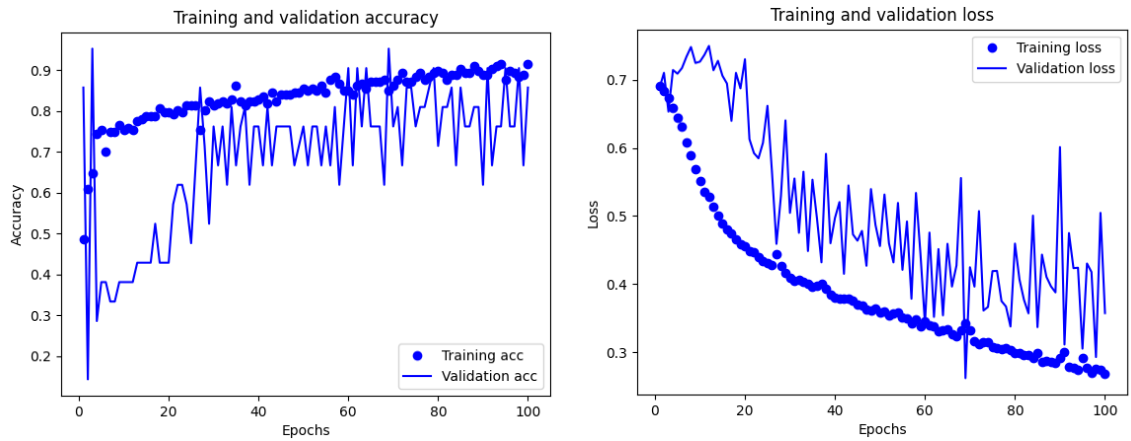


Рисунок 1, 2 – Графики точности и ошибок модели при одном промежуточном слое из 60 нейронов

При уменьшении количества нейронов в два раза можем наблюдать, что графики точности и потерь не сильно отличаются. Таким образом, можно сделать вывод, что размер слоя в 60 нейронов в первой модели был избыточен.

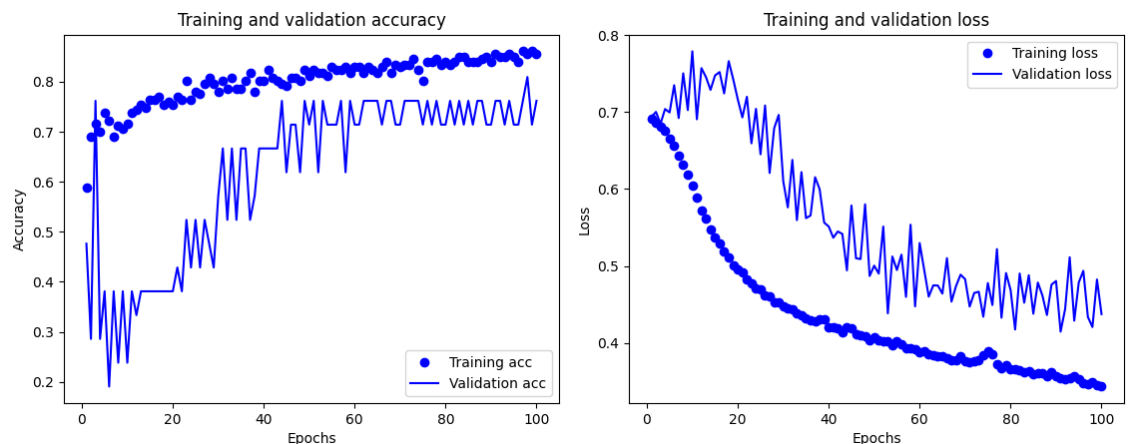


Рисунок 3, 4 – Графики точности и ошибки модели при одном промежуточном слое из 30 нейронов

Добавим промежуточный слой из 15 нейронов. Может наблюдать, как точность немного увечилась, а ошибки стали меньше. Таким образом, добавив второй слой мы смогли рассматривать комбинации изначальных признаков, что привело к большей точности.

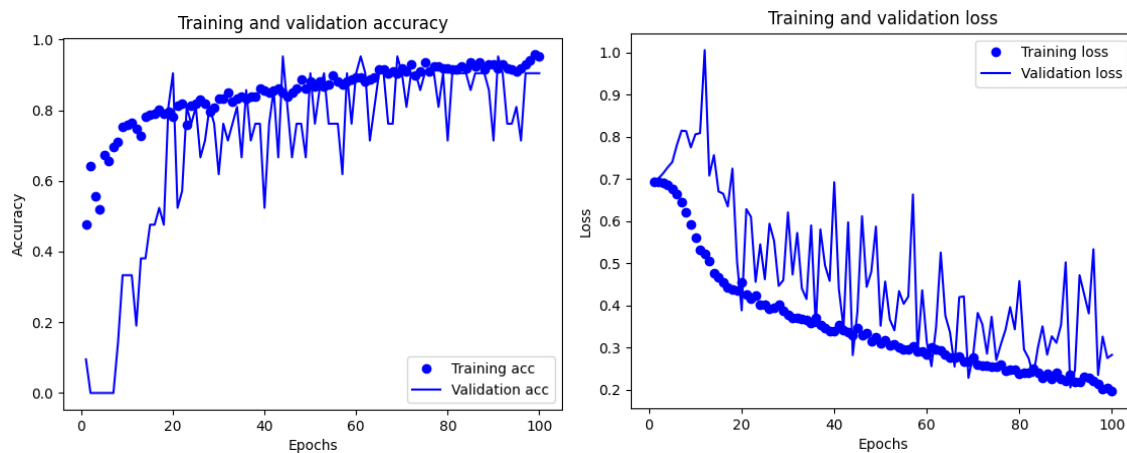


Рисунок 5, 6 – Графики точности и ошибки модели при двух промежуточных слоях из 30 и 15 нейронов соответственно

Выводы.

В ходе выполнения лабораторной работы было изучено влияние количества нейронов и слоев на обучение модели.

ПРИЛОЖЕНИЕ А

ИСХОДНЫЙ КОД ПРОГРАММЫ

```
import pandas
from tensorflow.keras.layers import Dense
from tensorflow.keras.models import Sequential
from tensorflow.keras.utils import to_categorical
from sklearn.preprocessing import LabelEncoder
import matplotlib.pyplot as plt
import matplotlib.colors as mclr
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, kernel_initializer='normal',
activation='relu'))
model.add(Dense(15, kernel_initializer='normal', activation='relu'))
model.add(Dense(1, kernel_initializer='normal', 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)
loss = H.history['loss']
val_loss = H.history['val_loss']
acc = H.history['acc']
val_acc = H.history['val_acc']
epochs = range(1, len(loss) + 1)
#Построение графика ошибки
plt.plot(epochs, loss, 'bo', label='Training loss')
plt.plot(epochs, val_loss, 'b', label='Validation loss')
plt.title('Training and validation loss')
plt.xlabel('Epochs')
plt.ylabel('Loss')
plt.legend()
plt.show()
#Построение графика точности
plt.clf()
plt.plot(epochs, acc, 'bo', 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()
```