|  |  |
| --- | --- |
| Picture 1 | МИНОБРНАУКИ РОССИИ  федеральное государственное бюджетное образовательное учреждение  высшего образования  «Санкт-Петербургский государственный морской технический университет» (СПбГМТУ) |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Факультет цифровых промышленных технологий

Направление подготовки 09.03.01

"**Интеллектуальные технологии киберфизических систем**"

Лабораторная работа №2

Вариант №15

Студент 2 курса группы 20221

Очного отделения

Руденко Вячеслав Сергеевич

Проверил:

Преподаватель CПбГМТУ

Кайнова Татьяна Денисовна

Санкт-Петербург

2025

СОДЕРЖАНИЕ

[ОБЩАЯ ЧАСТЬ 3](#_Toc191287272)

1. [1.1 Цель работы 3](#_Toc191287273)

[1.2 Формулировказадачи 3](#_Toc191287274)

[ХОД РАБОТЫ 4](#_Toc191287275)

[ЗАКЛЮЧЕНИЕ 5](#_Toc191287276)

[Приложение А 7](#_Toc191287277)

# ОБЩАЯ ЧАСТЬ

## Цель работы

Изучение методов прогнозирования цен на жильё с использованием глубоких нейронных сетей. Анализ влияния различных архитектурных изменений модели (добавление регуляризации, dropout и увеличение числа нейронов) на качество предсказаний.

## Формулировка задачи

* Исследовать работу базовой нейросетевой модели для прогнозирования цен на жильё на основе набора данных boston\_housing из библиотеки Keras.
* Изменить структуру модели, добавляя регуляризацию L2 и dropout, а также изменяя количество нейронов в скрытых слоях.
* Обучить все модифицированные модели и зафиксировать их ошибки на обучающей и тестовой выборках.
* Провести сравнительный анализ полученных моделей и сделать выводы о влиянии различных архитектурных решений на точность предсказаний.ХОД РАБОТЫ

# ЗАКЛЮЧЕНИЕ

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. ГОСТ Р 7.0.97-2016. Национальный стандарт Российской Федерации. Система стандартов по информации, библиотечному и издательскому делу. Организационно-распорядительная документация. Требования к оформлению документов: утвержден и введен в действие Приказом Федерального агентства по техническому регулированию и метрологии от 14.05.2018 N 244-ст: Дата введения 2018-07-01. - URL: <https://docs.cntd.ru/document/1200159234> (дата обращения: 02.10.2024). - Текст: электронный.

# Приложение А

Листинг кода

Листинг 1 - index.py

import tensorflow as tf

from tensorflow import keras

from tensorflow.keras.models import Sequential

from tensorflow.keras.layers import Dense, Dropout

from tensorflow.keras.regularizers import l2

import numpy as np

import pandas as pd

# Параметры обучения

EPOCHS = 500

BATCH\_SIZE = 16

# Загрузка данных

boston\_housing = keras.datasets.boston\_housing

(raw\_x\_train, y\_train), (raw\_x\_test, y\_test) = boston\_housing.load\_data()

# Стандартизация данных

x\_mean = np.mean(raw\_x\_train, axis=0)

x\_stddev = np.std(raw\_x\_train, axis=0)

x\_train = (raw\_x\_train - x\_mean) / x\_stddev

x\_test = (raw\_x\_test - x\_mean) / x\_stddev

# Конфигурации моделей

configs = {

    "Base Model": lambda: Sequential(

        [

            Dense(64, activation="relu", input\_shape=[13]),

            Dense(64, activation="relu"),

            Dense(1, activation="linear"),

        ]

    ),

    "L2 Regularization (0.1)": lambda: Sequential(

        [

            Dense(64, activation="relu", input\_shape=[13], bias\_regularizer=l2(0.1)),

            Dense(64, activation="relu", bias\_regularizer=l2(0.1)),

            Dense(1, activation="linear"),

        ]

    ),

    "L2 + Dropout (0.2)": lambda: Sequential(

        [

            Dense(64, activation="relu", input\_shape=[13], bias\_regularizer=l2(0.1)),

            Dropout(0.2),

            Dense(64, activation="relu", bias\_regularizer=l2(0.1)),

            Dense(1, activation="linear"),

        ]

    ),

    "More Neurons (128)": lambda: Sequential(

        [

            Dense(128, activation="relu", input\_shape=[13], bias\_regularizer=l2(0.1)),

            Dense(128, activation="relu", bias\_regularizer=l2(0.1)),

            Dense(1, activation="linear"),

        ]

    ),

    "Final Model (Dropout 0.3)": lambda: Sequential(

        [

            Dense(128, activation="relu", input\_shape=[13], bias\_regularizer=l2(0.1)),

            Dense(128, activation="relu", bias\_regularizer=l2(0.1)),

            Dropout(0.3),

            Dense(1, activation="linear"),

        ]

    ),

}

# Запуск всех конфигураций

for name, model\_fn in configs.items():

    model = model\_fn()

    model.compile(

        loss="mean\_squared\_error", optimizer="adam", metrics=["mean\_absolute\_error"]

    )

    history = model.fit(

        x\_train,

        y\_train,

        validation\_data=(x\_test, y\_test),

        epochs=EPOCHS,

        batch\_size=BATCH\_SIZE,

        verbose=0,

        shuffle=True,

    )

    # Получение финальных ошибок

    final\_train\_loss = history.history["loss"][-1]

    final\_val\_loss = history.history["val\_loss"][-1]

    # Вывод данных в нужном формате

    print(

        f"{name}: Ошибка обучения = {final\_train\_loss:.4f}, Ошибка теста = {final\_val\_loss:.4f}"

    )