

## Лабораторная работа 7. ПОЛНОСВЯЗНЫЕ НЕЙРОННЫЕ СЕТИ (FCNN). РЕШЕНИЕ ЗАДАЧ РЕГРЕССИИ И КЛАССИФИКАЦИИ

### 1. Изучение примеров. Изучите примеры:

[Lab\\_7\\_Ex1\\_Base+TensorFlow&Keras.ipynb](#), [Lab7\\_Ex2\\_MLP\\_Scikit-learn.ipynb](#)

### 2. Загрузка и подготовка данных.

2.1. В соответствии с индивидуальным вариантом загрузите предобработанные датасеты в формате CSV для решения задач регрессии и классификации.

2.2. К данным примените кросс-валидацию **k-fold**.

2.3. Выполните оптимизацию признакового пространства в датасетах.

### 3. Решение задачи регрессии и классификации с помощью FCNN.

3.1. Используйте FCNN (MLP) из библиотеки Scikit-learn.

3.1.1. Подберите гиперпараметры для алгоритма нейронной сети (НС) тремя способами: **Optuna**, **RandomizedSearchCV**, **Hyperopt**<sup>1,2</sup>.

3.1.2. Используйте оптимизаторы: **Adam**, **SGD** (стохастический градиентный спуск), **lbfgs** (оптимизация с использованием алгоритма BFGS).

3.2. Используйте FCNN посредством API Keras<sup>3,4</sup> и фреймворка TensorFlow.

3.2.1. Подберите гиперпараметры для алгоритма НС тремя способами: **Optuna**, **KerasTuner**<sup>5</sup>, **Ray Tune**<sup>6</sup>.

3.2.2. Используйте оптимизаторы: **Adam**, **SGD** (стохастический градиентный спуск), **RMSprop**<sup>7</sup>.

### 4. Визуализация.

4.1. Выведите архитектуры созданных НС.

4.2. Выведите графики обучения моделей (график потерь, график точности по эпохам).

**5. Оценка качества моделей.** Вычислите значения метрик оценки качества для обученных моделей регрессии и классификации. Полученные значения отобразите в табличной форме для задачи регрессии (см. образец). Для задачи классификации следует построить аналогичную таблицу.

---

<sup>1</sup> Hyperopt. – URL: <http://hyperopt.github.io/hyperopt/>

<sup>2</sup> Hyperopt. – URL: <https://github.com/hyperopt/hyperopt-sklearn>

<sup>3</sup> Библиотека Keras – Русскоязычная документация Keras. – URL: <https://ru-keras.com/home/>

<sup>4</sup> Keras. Deep Learning. – URL: <https://keras.io/>

<sup>5</sup> KerasTuner. – URL: [https://keras.io/keras\\_tuner/](https://keras.io/keras_tuner/)

<sup>6</sup> Ray Tune. – URL: <https://docs.ray.io/en/latest/tune/index.html>

<sup>7</sup> **RMSprop (Root Mean Squared Propagation)** – это алгоритм оптимизации, используемый в глубоком обучении и других методах ML для ускорения сходимости градиентного спуска. Он адаптирует скорость обучения для каждого параметра на основе исторической информации о градиентах

Регрессор/ Фреймворк	Train Data					Test Data				
	R <sup>2</sup>	MSE	MAE	Кол-во эпох	Оптимизатор	R <sup>2</sup>	MSE	MAE	Кол-во эпох	Оптимизатор
	0.XX	0.XXXX	0.XXXX			0.XX	0.XXXX		0.XXXX	0.XXXX
...										

**6. Деплой модели.** Создайте по одному запросу с целью получения прогноза для задачи регрессии и задачи классификации (см. [Lab7\\_Ex2\\_MLP\\_Scikit-learn.ipynb](#)).

**7. Вывод.** Напишите вывод о выполненной *Лабораторной работе №7*, в котором выберите лучшую модель регрессии и классификации, реализованные с помощью FCNN, и обоснуйте свое решение. Сравните полученные результаты с моделями ML, полученными в предыдущих Лабораторных работах.