

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

**ОТЧЕТ**  
**по практической работе №5**  
**по дисциплине «Искусственные нейронные сети»**  
**Тема: Оптимизация нейронных сетей в библиотеке Keras**

Студентка гр. 8383

\_\_\_\_\_

Максимова А.А.

Преподаватель

\_\_\_\_\_

Жангиров Т.Р.

Санкт-Петербург

2021

## Цель работы

Необходимо в зависимости от варианта сгенерировать датасет и сохранить его в формате csv. Построить модель, которая будет содержать в себе автокодировщик и регрессионную модель. Схематично это должно выглядеть следующим образом:



Обучить модель и разбить обученную модель на 3: Модель кодирования данных (Входные данные → Закодированные данные), модель декодирования данных (Закодированные данные → Декодированные данные) и регрессионная модель (Входные данные → Результат регрессии).

В качестве результата представить исходные код, сгенерированные данные в формате csv, кодированные и декодированные данные в формате csv, результат регрессии в формате csv (что должно быть и что выдает модель), и сами 3 модели в формате h5.

## Вариант 1

Цель регрессии:  $(27 \% 7) + 1 = 7$

$X \in N(3, 10)$

$e \in N(0, 0.3)$

Признак	1	2	3	4	5	6	7
Формула	$X^2 + e$	$\sin\left(\frac{X}{2}\right) + e$	$\cos(2X) + 2$	$X-3+e$	$-X + e$	$ X  + e$	$\frac{(X^3)}{4} + e$

## Основные теоретические положения

### Выполнение работы

#### 1. Генерация датасета

Была написана программа для генерации тестовых (2000 объектов) и контрольных (500 объектов) данных. Сгенерированные данные автоматически сохраняются в файлы с разрешением csv. Пример содержимого файла представлен ниже. Здесь с 1 по 6 столбец подаваемые в ИНС данные, 7 столбец - значения, которые нужно предсказать.

```
1 64719370875807e+01, 7.788478111652094872e-01, -8.484846712553318904e-01, -3.118438824295461778e+01, 8.138375995451165849e+00, 8.138375995451165849e+00, -1.3573817351
2 81480821578819e+01, 4.298251897599285444e-01, -1.280231102381984773e+00, 1.486771071482922641e+00, -5.078796884519478170e+00, 4.686771071482922864e+00, 2.8891597382897
3 89848759781271e+01, -1.861249242258123449e+00, -1.142925337293288823e+00, -8.782513889638816389e+00, 4.636869583414656937e+00, 4.636869583414656937e+00, -3.50737461138
4 67394974644037e+02, -8.236138527493744635e-01, 2.729259538056794110e-01, 7.245477965628956518e+00, -1.001601330912289889e+01, 1.024547796562895563e+01, 2.6804942284940
5 8015240829997e+01, 5.288539243848152811e-02, 9.672874213958525846e-01, -8.664646889829562260e+00, 6.144958827225289880e+00, 6.144958827225289880e+00, -5.1229986150598
6 63247987583670e+00, -4.162278822578747151e-01, -6.638461452777324467e-01, -4.107223897753517571e+00, 1.480880260291921856e+00, 1.480880260291921856e+00, -3.54814151588
7 43181380734978e+01, -8.328840495338287742e-01, -1.121625889991100666e+00, -7.823179218571738154e+00, 4.579437958346681611e+00, 4.579437958346681611e+00, -2.60993867861
```

2. После была написана программа, реализующая построение модели, содержащей автокодировщик и регрессионную модель. Данная модель принимает нормализованные данные и 1) возвращает предсказанное на их основе значение 2) возвращает декодированные значения входных данных.

3. Далее модель была разделена на 3 отдельные модели, согласно условиям задачи. Полученные модели были протестированы на тестовых данных (результаты сохранены), а после сохранены в формате h5.

#### 4. Результаты тестирования:

Исходные данные (7 столбиков):

```
6.651564719378075807e+01, 7.788478131652894572e-01, -8.484846712553318904e-01, -1.118435824295461778e+01, 8.130375995451165849e+00, 8.130375995451165849e+00, -1.3 ✓
2.318181400821578819e+01, 4.290251897599285444e-01, -1.200231102301904773e+00, 1.606771071482922641e+00, -5.078796084519470170e+00, 4.606771071482922641e+00, 2.80915
2.619289049759701271e+01, -1.061249242258123449e+00, -1.142925337293288823e+00, -8.782513889638016309e+00, 4.636869583414656937e+00, 4.636869583414656937e+00, -3.507
1.02746739497464037e+02, -8.236130527493744635e-01, 2.729259538056794110e-01, 7.245477965628956510e+00, -1.001601330912289889e+01, 1.024547796562895653e+01, 2.60049
```

Исходные данные в нормализованном виде (6 столбиков - так как 7 ответы):

```
# Нормализованные тестовые данные
-2.806319295465709573e-01, 1.012906985244152791e+00, -1.137730053496808225e+00, -1.144376715139228784e+00, 1.138438392913607267e+00, -4.129393002015492586e-02
-5.619553976285603936e-01, 5.570581968779486637e-01, -1.600617686865557276e+00, 1.386246290555401550e-01, -1.851643749218544188e-01, -6.027068074583210588e-01
-5.424074735613074738e-01, -1.384896807505817229e+00, -1.525205038429298554e+00, -8.954378166954979346e-01, 7.881653468226316184e-01, -5.979112379904556462e-01
-4.541949182694685755e-02, -1.075236534522285609e+00, 3.380123121743631165e-01, 7.042894454441930607e-01, -6.809918034547953747e-01, 2.957034090048346608e-01
```

Кодирование (8 столбиков):

```
# Закодированные данные - ТЕСТ
-6.842402219772338867e-01, 1.401645851135253906e+01, 4.632065773010253906e+00, -3.909359693527221680e+00, -9.581965446472167969e+00, 7.95158630676269531e+00, 7.2890195846
2.372917890548706055e+00, 7.242196559986808585e+00, 7.067207336425781250e+00, -3.158989667892456055e+00, -3.145221471786499023e+00, -1.529122114181518555e+00, -3.421956062
1.568351686000823975e-01, 9.597330093383789062e+00, 7.724462032318115234e+00, -5.185853958129882012e+00, -7.349640369415283203e+00, 2.710767745971679688e+00, 9.51266765594
1.291549015045166016e+01, 2.170157412556152344e+01, 6.602426052093505859e+00, 4.341302852630615234e+00, 6.644321441650390625e+00, -9.143837928771972656e+00, -1.23383350372
```

Декодирование кодированных данных (6 столбиков):

```
# Декодированные данные - ТЕСТ
-2.220442891120910645e-01, 1.167185068130493164e+00, -7.021838426589965820e-01, -1.136458396911621094e+00, 1.118741989135742188e+00, -3.627483546733856201e-02
-5.883764624595642090e-01, 2.867740392684936523e-01, -1.807123184204101562e+00, 9.439127147197723389e-02, -2.174543589353561401e-01, -7.236369252204895020e-01
-5.835912227630615234e-01, -1.532883405685424805e+00, -1.345339536666870117e+00, -8.538377285003662189e-01, 7.511378526687622070e-01, -5.425650477409362793e-01
-4.211624339222908020e-02, -9.71383571624758594e-01, 3.712525963783264160e-02, 6.978078901878727539e-01, -7.495163679122924805e-01, 2.830466628074645996e-01
```

Регрессия (ожидаемое и полученное значения):

```
# Должно быть/Предсказала модель
-1.357301735827010418e+02, -1.451972351074218750e+02
2.809159730289776746e+01, 2.781748962402343750e+01
-3.507374611389820984e+01, -3.671990585327148438e+01
2.600494220494088040e+02, 2.582880859375000000e+02
```

#### Вывод

Как видно, во многих случаях исходные нормализованные данные совпадают с декодированными данными, а ожидаемые значения регрессии совпадают с исходными (правильными) значениями.