

Вариант 4

Цель регрессии 1 (номер зачетки 7)

$$X \in N(0,10)$$

$$e \in N(0,0.3)$$

Признак	1	2	3	4	5	6	7
Формула	$\cos(X)+e$	$-X+e$	$\sin(X)*X+e$	$\sqrt{ X }+e$	X^2+e	$- X +4$	$X-(X^2)/5+e$

Генерация данных

Для генерации данных реализована функция `generate_data`. Результатом ее работы является создание четырех файлов формата csv: обучающий и тестовый наборы входных данных, а также обучающий и тестовый наборы выходных данных. Обучающие и тестовые данные состоят из 500 и 100 строк соответственно.

Сначала генерируются величины X и e , являющиеся выборками из нормальных распределений с указанными параметрами.

Входные данные состоят из величин, полученных при выполнении над сгенерированными выборками формул 2-7, выходные данные являются результатом применения к этим выборкам формулы 1.

При последующей загрузке данных в программу они нормализуются для упрощения обучения нейронной сети.

Построение модели

В работе создается модель нейронной сети с одним входом и двумя выходами, имеющая следующую структуру, приведенную на рис. 1.

Модель компилируется с оптимизатором Adam, в качестве функции потерь используется среднеквадратичная ошибка (MSE). Потери регрессии и декодирования учитываются с весами 1 и 0.8, так как эмпирическим путем было установлено, что при одинаковых весах декодирование осуществляется точнее. Обучение модели осуществляется в течение 100 эпох, батчами по 50 образцов.

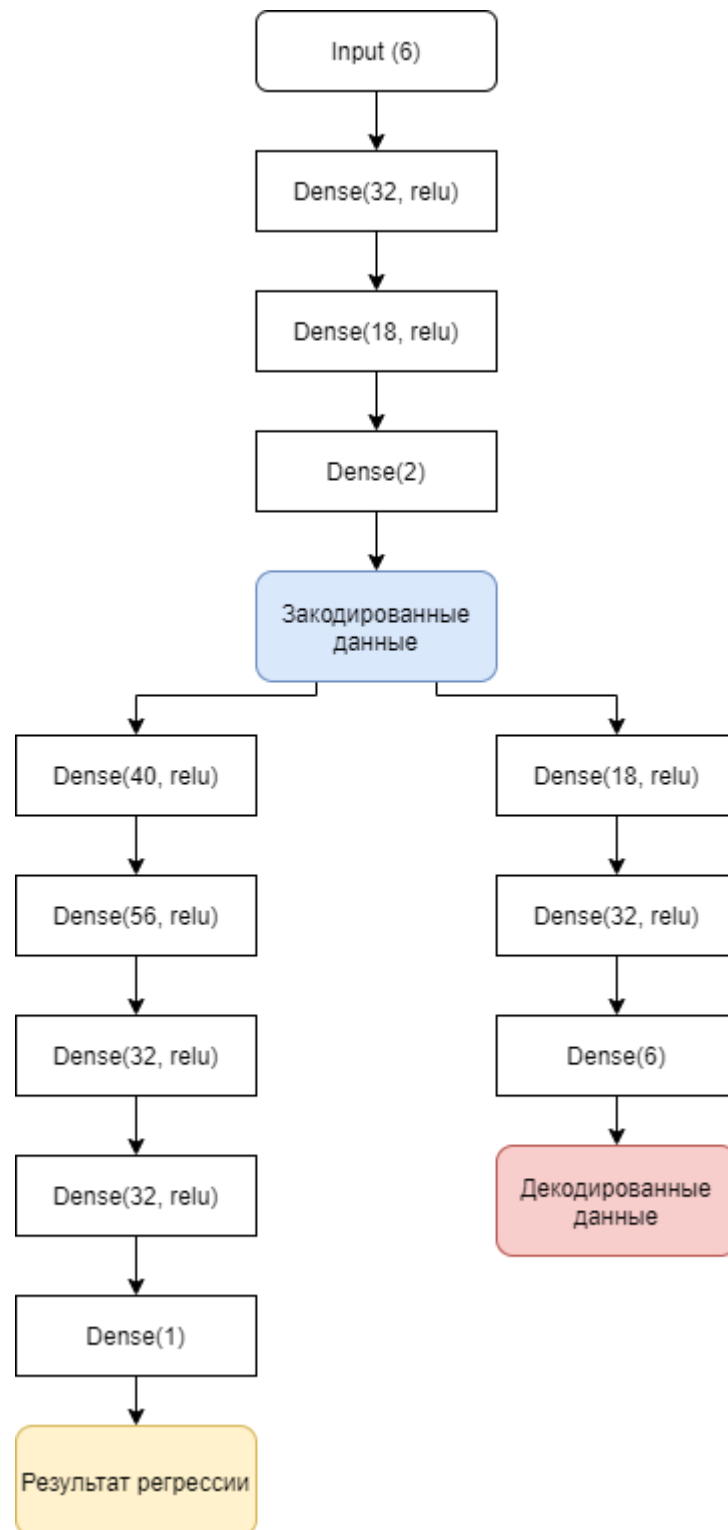


Рисунок 1

На тестовом множестве была оценена точность модели:

```

4/4 [=====] - 0s 3ms/step - loss:
0.0943 - regression_loss: 0.0700 - decoding_loss: 0.0304 -
regression_mae: 0.1214 - decoding_mae: 0.1124
  
```

Полученные значения означают, что средняя разница между реальным значением и результатом регрессии составляет 0.1214, а между исходным

значением и значением, полученным после кодирования и декодирования, – 0.1124.

Далее полученная модель разделяется на три части: модель кодирования данных, модель декодирования и регрессионную модель. Модель кодирования и регрессионная модель задаются входными и выходными слоями:

```
encoding_model = keras.Model(inputs, encoded)
regression_model = keras.Model(inputs, regression)
```

Для модели декодирования необходимо создать новый входной слой, по размеру соответствующий выходу модели кодирования. Последующие слои извлекаются из общей модели и вызываются с новыми входными данными:

```
encoded_input = keras.Input(shape=(2,))
decoding_layer = model.get_layer("dl_1")(encoded_input)
decoding_layer = model.get_layer("dl_2")(decoding_layer)
decoding_layer = model.get_layer("decoding")(decoding_layer)
decoding_model = keras.Model(encoded_input, decoding_layer)
```

Полученные модели сохраняются в формате h5.

Предсказание результатов

Получаем закодированные данные при помощи модели кодирования:

```
encoded_data = encoding_model.predict(test_data)
```

При помощи модели декодирования из закодированных данных получаем декодированные:

```
decoded_data = decoding_model.predict(encoded_data)
```

Так как исходные наборы данных были нормализованы перед их обработкой нейросетью, результат декодирования также будет нормализован. Для удобства проверки декодированные данные приведены как в нормализованном, так и в денормализованном виде. Тестовый датасет также был дополнительно приведен в нормализованном виде.

В таблице ниже приведены данные из первых строк декодированного и тестового наборов данных в нормализованном виде.

Исходные данные	Декодированные данные
-0.7368978403089843088	-0.6795078516006469727

1.023857258148376248	0.7126291990280151367
-0.1323971140495812937	0.03686447441577911377
-0.3121332083512891042	-0.4899392724037170410
0.07859215916312911610	0.2571344673633575439
0.5063987629588102557	0.6229287385940551758

С помощью регрессионной модели получаем предсказания значений первой формулы на основе значений остальных шести:

```
regression_predictions = regression_model.predict(test_data)
```

Результаты предсказаний записываются в файл csv вместе с реальными значениями. Ниже приведены первые шесть строк полученного набора данных.

Реальные значения	Предсказанные значения
-0.06991432048274237587	-0.1152272522449493408
-1.196014051375988707	-1.097622394561767578
-1.546049402145680762	-1.286071777343750000
0.9929434385378691497	0.9733257293701171875
0.8423446132138675546	0.8116227388381958008
0.5142868786660669267	0.4457509219646453857