

Практическое задания 5

Вариант 1

Цель регрессии – 3 признак

Вариант 1

$X \in N(3, 10)$
 $e \in N(0, 0.3)$

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

Генерация данных:

```
def genData(self, size):  
    x_data = []  
    y_data = []  
    for i in range(size):  
        X = np.random.normal(3, 10)  
        e = np.random.normal(0, 0.3)
```

Данные генерируются с помощью функции `genData()`, которая принимает количество строк, и возвращает массив.

Генерируются данные для обучения и для проверки. Сгенерированные данные сохраняются в файлы `data_x.csv` и `data_y.csv`

Описание 7 признаков:

Каждый признак был описан в отдельной функции, каждая из которых будет вызываться дальше.

```
def fun_1(self, X, e):  
    return X**2 + e  
  
def fun_2(self, X, e):  
    return math.sin(X/2)+e  
  
def aim_fun(self, X, e):  
    return math.cos(2*X)+e  
  
def fun_4(self, X, e):  
    return X-3+e  
  
def fun_5(self, X, e):  
    return -X+e  
  
def fun_6(self, X, e):  
    return math.fabs(X)+e  
  
def fun_7(self, X, e):  
    return (X**3)/4 + e
```

Создание модели:

Создание модели были разделены на три части: Энкодер, Декодер и Регрессия.

```

encoded = Dense(32, activation='tanh')(input)
encoded = Dense(32, activation='relu')(encoded)
encoded = Dense(32, activation='tanh')(encoded)
encoded = Dense(6, activation='tanh')(encoded)

decoded = Dense(6, activation='tanh')(encoded)
decoded = Dense(32, activation='relu')(decoded)
decoded = Dense(32, activation='tanh')(decoded)
decoded = Dense(6, activation='linear', name='decoder_output')(decoded)

regr = Dense(64, activation='relu')(encoded)
regr = Dense(64, activation='tanh')(regr)
regr = Dense(64, activation='tanh')(regr)
regr = Dense(64, activation='tanh')(regr)
regr = Dense(32, activation='tanh')(regr)
regr = Dense(1, activation='linear', name="regr_output")(regr)

```

Модель имеет два выхода: предсказание регрессии и декодированные параметры .

Модель имеет по 5 скрытых слоев для кодирования и декодирования и 4 слоя для регрессии.

Параметры обучения модели:

- Оптимизатор: adam,
- Функция потерь: mean squared error
- Число эпох: 60,
- Batch_size: 6

Результаты к 60 эпохе:

Epoch 60/60

loss: 0.3094 - regr_output_loss: 0.2905 - decoder_output_loss: 0.0189 - regr_output_mae: 0.4093 - decoder_output_mae: 0.0950

На тестовых данных:

val_loss: 1.0195 - val_regr_output_loss: 0.9963 - val_decoder_output_loss: 0.0232 - val_regr_output_mae: 0.7737 - val_decoder_output_mae: 0.1103

